

CONTRACT: NAS 7-1070

**EOS WORKSTATION
PHASE-II SBIR FINAL REPORT**

Performance Period June 2, 1989 through January 3, 1992

Franz Leberl

Milan Karspeck

Michel Millot

Kelly Maurice

Matt Jackson

**VEXCEL Corporation
2477 55th Street, Suite 201
Boulder, CO 80301**

7 April 1992

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

1. Report No. Final Report		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle EOS Workstation -- Final Report				5. Report Date April 7, 1992	
				6. Performing Organization Code	
7. Author(s) Franz Leberl, Milan Karspeck, Michel Millot, Kelly Maurice, Matt Jackson				8. Performing Organization Report No.	
				10. Work Unit No. NAS7-1070	
9. Performing Organization Name and Address VEXCEL Corporation 2477 55th Street, Suite 201 Boulder, CO 80301				11. Contract or Grant No.	
				13. Type of Report and Period Covered Final Report	
12. Sponsoring Agency Name and Address National Aeronautics & Space Administration Washington, DC 20546-0001 NASA Resident Office - Jet Propulsion Laboratory 4800 Oak Grove Drive, Pasadena, CA 91109				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract This final report summarizes the work done from mid-1989 until January 1992 to develop a prototype set of tools for the analysis of EOS-type images. Such images are characterized by great multiplicity and quantity. A single "snapshot" of EOS-type imagery may contain several hundred component images so that on a particular pixel, one finds multiple gray values. A prototype EOS-sensor, AVIRIS, has 224 gray values at each pixel. The work focused on the ability to utilize very large images and continuously roam through those images, zoom and be able to hold more than one black and white or color image, for example for stereo viewing or for image comparisons. A second focus was the utilization of so-called "image cubes," where multiple images need to be co-registered and then jointly analyzed, viewed and manipulated. The target computer platform that was selected was a high-performance graphics superworkstation, Stardent 3000. This particular platform offered many particular graphics tools such as the Application Visualization System (AVS) or Doré, but it missed availability of commercial third-party software for relational data bases, image processing, etc. The project was able to cope with these limitations and a Phase-III activity is currently being negotiated to port the software and enhance it for use with a novel graphics superworkstation to be introduced into the market in the Spring of 1993.					
17. Key Words (Suggested by Author(s)) Image cubes, digital light table, stereoscopy, image co-registration, radar image analysis.				18. Distribution Statement	
19. Security Classif. (of this report)		20. Security Classif. (of this page)		21. No. of pages	
				22. Price	

13. STIMS 3X ACC# 9336071 IPS-FILE ADABAS # 186516
FICHE AVAIL = AD HARD COPY AVL = OK
ORIG AGENCY = NASA RECEIPT TYPE = REG
DOCUMENT CLASS= TRP ACCESS LEVEL = O
LIMITATION CAT= NONE DOCUMENT SEC = NC
SUBJECT CATGRY= 54 SPECIAL HANDL =
INC AUTHOR LST= N INC CNTRCT LST= N
COUNTRY ORIGIN= US COUNTRY FINANC= US
PUB DATE = 19920407 CORP SOURCE = V0987612
ABSTRACT PREP = NON

TITLE = EOS Workstation
TITLE SUPP = Final Report, 2 Jun. 1989 - 3 Jan. 1992
AUTHOR = LEBERL, FRANZ
AUTHOR = KARSPECK, MILAN
AUTHOR = MILLOT, MICHEL
AUTHOR = MAURICE, KELLY
AUTHOR = JACKSON, MATT
CONTRACT NUM = NAS7-1070
CONTRACT NUM = SBIR-07.07-0094
REPORT NUM = NASA-CR-191326
REPORT NUM = NAS 1.26:191326
SALES AGY PRIC= Contact the JPL SBIR Field Center Manager for furt
SALES AGY PRIC= her information
MAJOR TERMS = COMPUTER GRAPHICS
MAJOR TERMS = DATA BASES
MAJOR TERMS = EARTH OBSERVING SYSTEM (EOS)
MAJOR TERMS = IMAGE PROCESSING
MAJOR TERMS = SOFTWARE TOOLS
MAJOR TERMS = STEREOSCOPY
MAJOR TERMS = WORKSTATIONS
MAJOR TERMS = IMAGING TECHNIQUES
MINOR TERMS = RADAR IMAGERY
MINOR TERMS = REMOTE SENSING
MINOR TERMS = SATELLITE IMAGERY
MINOR TERMS = SUPERCOMPUTERS
FORM OF INPUT = HC

***** END OF ADABAS RECORD # 186516 *****

CONTENTS

	Page
EXECUTIVE SUMMARY	1
CHAPTER 1. INTRODUCTION	4
CHAPTER 2. CURRENT STATUS OF EXISTING REMOTE SENSING ANALYSIS SOFTWARE	13
CHAPTER 3. FUNCTIONAL REQUIREMENTS	18
CHAPTER 4. SELECTION OF A TARGET PLATFORM FOR AN EOS-TYPE WORKSTATION	21
CHAPTER 5. EOS WORKSTATION SPECIFICATIONS	25
CHAPTER 6. ISSUES CONCERNING THIRD-PARTY SOFTWARE	38
CHAPTER 7. INTERACTING WITH LARGE OVERLAPPING STEREO IMAGES: THE DIGITAL LIGHT TABLE LT	44
CHAPTER 8. CREATION OF, AND INTERACTION WITH, IMAGE CUBES	47
CHAPTER 9. STATUS OF THE EOS WORKSTATION AND TRANSITION TO PHASE-III	53
CHAPTER 10. RECOMMENDATIONS AND OUTLOOK	56
REFERENCES	58
ACKNOWLEDGEMENT	59



EXECUTIVE SUMMARY

This document represents the final report of a Small Business Innovation Research (SBIR) Phase-II effort to develop a software system on a computer target workstation to support remote sensing research in the era of NASA's Earth Observing System (EOS). The combination of the target platform and software are denoted as "EOS Workstation." The purpose of the software is to support scientific users of EOS-type data. Such data consist at the current time of experimental data flown on a NASA aircraft and data obtained from the Space Shuttle or current remote sensing satellites.

A two and one-half year effort began in June of 1989 to review, refine and implement a design that was presented in an SBIR Phase-I project preceding the current effort.

The work was initiated by embarking on a refined definition of the functions needed for EOS-type research, resulting in a Functional Requirements Document for the EOS workstation. Upon completion of the functional requirements document, a set of parallel efforts were begun to have a system prototype built. These included:

- a. definition of a target platform for high performance computing of massive amounts of image data;
- b. examination of a variety of existing public domain and third-party software packages that can be integrated into the EOS platform;
- c. collection of demonstration data sets to demonstrate certain functionalities of the EOS workstation;
- d. definition of a software engineering project beginning with the creation of an analysis document and subsequent implementation of software.

It was the purpose of the effort to create a prototype software system executable on a defined target computer platform. This goal has been accomplished and an elaborate software system has been created under this effort that can be demonstrated to successfully process a variety of EOS-type image data to include:

- a. high resolution AVIRIS data;
- b. AIRSAR synthetic aperture radar images;
- c. LANDSAT thematic Mapper data;
- d. SPOT panchromatic and multispectral images;
- e. European Remote Sensing Satellite ERS-1 radar images;
- f. a variety of Geographic Information System data from the U.S. Geological Survey and other sources of map data.

A major difficulty of the effort resulted from the choice of target platform. Early in the project a new computer system was selected which offered particular promise in processing multi images kept in a so-called "image-cube," and it appeared to support the visualization and interaction with images using stereoscopy under the X-Windows Graphical User Interface (GUI).

Upon selection, delivery and installation of this target platform, it became slowly apparent that the third-party software needed as part of an integrated EOS workstation would not easily be ported to this new and exciting computer. The fact emerged that no third-party commercial off-the-shelf Geographical Information System package, relational data base package, generic image processing package nor visualization software would be ported to this computer by other parties. Therefore, an inordinate amount of work had to be added in this project which was originally not expected to be expended to obtain a basic infrastructure of capabilities which on other platforms may have been available.

However, there is also a positive side for this innovative computer architecture: it permitted one to implement very new and innovative means of interacting with multiple

images. As we will show in this report, the use of stereoscopy and the analysis of so-called image-cubes was very well supported on this platform and these capabilities are now demonstrated and available for Phase-III work.

We can also report that the transition from Phase-II to Phase-III is taking shape, particularly because of the choice of platform that was made. The current vendor of that platform is Kubota Pacific Computer Corporation, a subsidiary of a Japanese concern. This company has signed on as a "partner" to exploit the exciting new Alpha chip of Digital Equipment Corporation of Maynard, Massachusetts. It appears that the software done under Phase-II will directly feed into a new and innovative imaging computer that Kubota is currently building in collaboration with Digital Equipment Corporation.

CHAPTER 1

INTRODUCTION

1.1 Background and Objectives

With the advent of sensing technologies capable of producing multiple images during one pass of the sensor over the Earth's surface, one is left with a need to cope with very large data quantities. The example of AVIRIS produces an image consisting of 1,000 by 1,000 pixels and at each pixel 224 gray values. At a resolution on the ground of 10 m per pixel and an aircraft speed of 200 m per second, this data set is being produced in an elapsed time of 50 seconds, resulting in 224 Mbytes of image data.

The AIRSAR is a synthetic aperture radar flown aboard a DC-8 aircraft by the Jet Propulsion Laboratory and produces in one pass at each pixel 15 gray values, namely the five values of the Stokes matrix described in the polarization of the reflected radar signal and the sensor does this for three frequency bands simultaneously, namely C-, L-, and P-band. Again 15 images are produced simultaneously in one single pass when previous technologies of remote sensing typically did not produce more than one to four images in one pass.

In addition the requirement is emerging to perform remote sensing work at very specific times during the year or during the diurnal cycle between day and night. As a result a particular study area may be imaged many times under different conditions resulting in multiple data sets of which each component may be a massive data set by itself when compared with conventional standards of remote sensing.

Finally, the existing knowledge base about a certain area or region of interest is a result of perhaps a century of diligent work by a range of geoscience disciplines, to include geological maps, forestry maps, land-use maps, hydrological data, and of course the conventional topographic information that is captured in national programs and

1

standardized mapping products. These existing data must not be ignored in any analysis of new remote sensing imagery. Therefore, a requirement emerges to merge together into an analysis system the new and timely imagery with the known information about the environment.

The current project was driven to cope with these new requirements of dealing, on one hand, with massive amounts of existing map data and, on the other hand, with the massive amounts of new remote sensing imagery. Clearly no existing solutions were available at the time to cope with the massive amounts of data that a scientific user of remote sensing images is confronted with.

Remote sensing very clearly has become very data-driven and the analysis tools lag behind. The Earth Observing System's massive expected data flow fortunately will not reach the scientific user until the end of this decennium [or the beginning of the next millennium]. In the meantime it will be necessary to develop the concepts and software that will allow the human interpreter to cope with these massive data sets. Also in the meantime one will need to develop an understanding of the usefulness of the data and various analysis concepts by simulation and experimental data acquisition aboard research aircraft and from prototype sensors on those aircraft.

The task at hand is, therefore, to build the prototype software system which would support the study and analysis of innovative remote sensing data sets. Two broad and competing objectives can be defined for this work, namely

- a. the creation of a useful and very straightforward software tool box to deal with known analysis concepts and implementing those on a platform that is currently the favorite among scientific users; or

- b. the implementation of innovative and previously unavailable analysis concepts, whereby a certain amount of trial and error is permitted, to put together a prototype software system that will allow the scientific user to experiment with these new analysis concepts.

We opted for the second approach which presents the higher risk. And in the process we did not strive to implement traditional analysis concepts on a standard computer such as a personal computer, Macintosh or low cost RISC-workstation.

Given the dramatic innovation cycles in computer hardware, the project took to a certain disregard for the "platform du jour" and emphasized analysis concepts whether they were inexpensively supported by current hardware or not.

In a formal manner the work was based on three technical objectives as defined in the proposal for the SBIR Phase-II work. These are:

- * Technical Objective #1 -- Implementation/Integration of Major Software Organization Modules on a Proposed Hardware System Architecture.
- * Technical Objective #2 -- Implementation of Key Applications Modules that are Relevant to EOS and Pre-EOS Remote Sensing Programs, particularly for geometric image processing and image data visualization.
- * Technical Objective #3 -- Demonstration of the flexibility of the software system design by integrating selected modules from third-party application software packages.

These objectives were maintained throughout the work and were accomplished. Some changes occurred along the way as a result of the very rapid innovation cycles and computer architectures. One might add a Technical Objective #4 which evolved from the initial work under this Phase-II program, namely

- * Technical Object #4 -- Experiment with innovative concepts for the interaction between the user and extensive multiple-image data sets (image cube).

1.2 Summary

A prototype image analysis workstation for scientists dealing with EOS-type remote sensing images has been developed. This prototype software should be useful for preparatory studies of aircraft data that have EOS-type characteristics. These include high-resolution spectrometer data from the AVIRIS sensor and multipolarization, multifrequency synthetic aperture radar (SAR) data from sensors that are currently flown on a NASA-operated aircraft.

The system is also capable of processing current satellite remote sensing data, such as that from LANDSAT, SPOT and ERS-1. The focus is on integrating capabilities that are common to a large variety of remote sensing studies with modern aircraft and satellite imagery.

1.2.1 Functionality

The purpose of this workstation is to provide EOS investigators with a versatile tool for rectifying, co-registering, and geocoding images from sensors in the pre-EOS era, for visualizing these images and their interrelationships, and for extracting features and interpretive data from the images for subsequent import to a Geographic Information System (GIS). It also allows them to manipulate non-image information (spatial features, spectral or polarization signatures). Various types of imagery can be integrated via image cubes consisting of co-registered images from different sensors or spectral bands. The image cubes and non-image objects extracted from the image cubes are available to other systems such as user-specific software and GIS (see Figure 1.1).

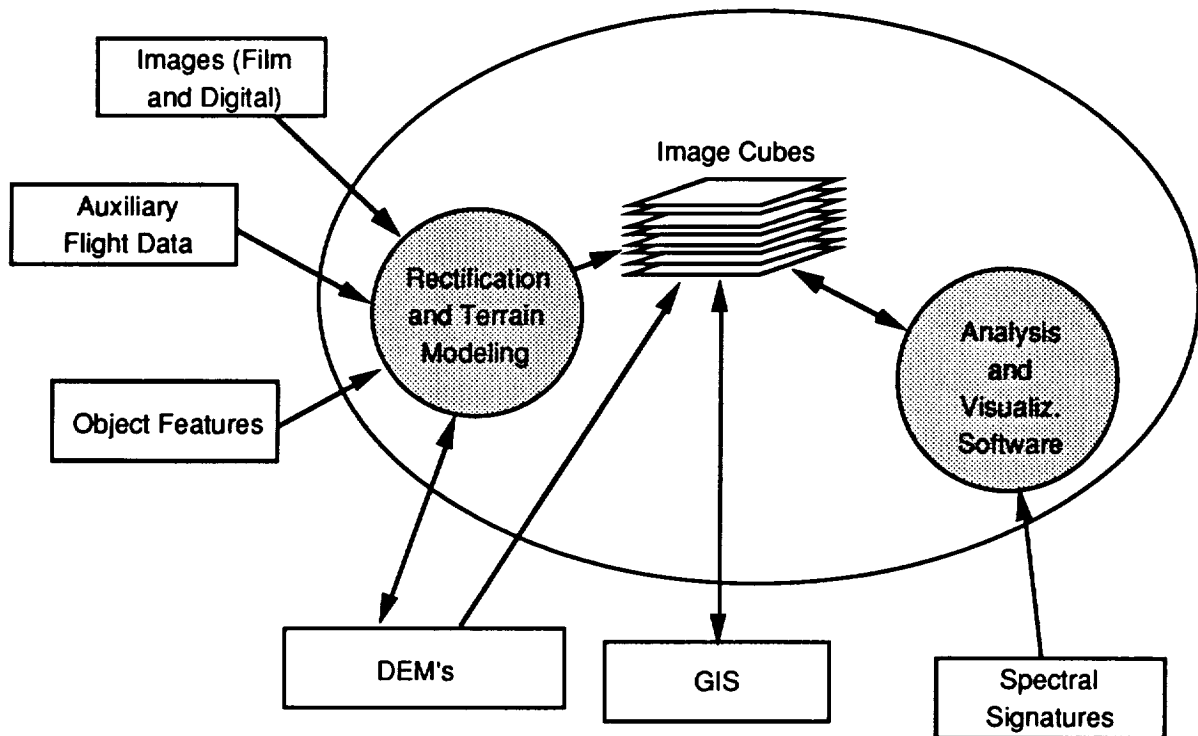


Figure 1.1: Overall concept of the EOS-Workstation.

1.2.2 Sensors

The EOS workstation is capable of processing the following remote sensing image data:

- LANDSAT thematic mapper;
- SPOT color and panchromatic imager;
- AVIRIS aircraft sensor images;
- multipolarization, multifrequency SAR from aircraft;
- digitized aerial photography;

The EOS workstation is also capable of working with digital elevation models, digital map data from a GIS, ground truth data such as spectral and other object signatures (backscatter, etc.), and various auxiliary flight data that accompany the images.

1.2.3 Initial Target Hardware

The initial implementation of this software system is on a high performance graphics supercomputer that can handle image data of high dimensionality and large coverage. This is a Stardent 3000VS computer with:

- 32 MFLOPS vector floating point processor;
- 16 MFLOPS scalar floating point processor;
- 32 MIPS integer processor;
- 32 MB main memory;
- 2.2 GB disk storage.

Stereo viewing of imagery and extracted data is on a 19" 1280 x 1024 color monitor equipped with a Tektronix SGS 625 liquid crystal shutter and polarized glasses.

1.2.4 Visualization Software

The workstation makes extensive use of Stardent's visualization software, including the Doré 3-D rendering library, the MATLAB data exploration program, and the AVS network editor. The latter allows a user to quickly prototype complete visualization applications by linking modules from a library of image processing and rendering routines -- all via a simple point-and-click window interface. The interactive, visual, and exploratory nature of these tools allows the capabilities of the workstation to evolve to meet the needs of the researchers during the pre-EOS study phase.

The project provides a base set of analysis and visualization capabilities, centered around the image and GIS data resident in the co-registered image cubes. These capabilities include real-time slicing through the spectral planes, perspective viewing of the terrain overlaid with texture-mapped images, multi-spectral classification, and tools to identify spectral prototypes. Also included is a "light table" program that allows a user

100

to easily roam and zoom large digital images in either monoscopic or stereoscopic mode.

1.2.5 Status of Software

The EOS workstation software has been developed to a point where it can be demonstrated with the support of a cognizant engineer. It represents a set of prototype capabilities which will need to be reprogrammed and stabilized for delivery to an end-user at a location different from the developing laboratory.

An initial application of many of the software modules developed for the EOS workstation is to process Magellan images taken from the Magellan radar orbiting around planet Venus. The capabilities particularly include:

- * monoscopic and stereoscopic viewing and measuring in a light table and comparator mode;
- * processing of digital elevation data with the digital elevation sub-modules (see Figure 1.1, Function Box DEMs);
- * application of the visualization software in a separate project for visualization and analysis of geologically relevant AVIRIS data (see Maurice et al., 1990).

As part of a Phase-III effort the software is currently being ported away from the originally selected target platform (the Stardent 3000) into an open environment to include computers by Silicon Graphics, Incorporated (SGI) and other computers. This port will ultimately lead to a Phase-III activity in collaboration with the successor company to the original Stardent manufacturer, namely Kubota Pacific Computer Corporation in Santa Clara, California. Availability of this software is expected to be in the first quarter of 1993.

1.3 Organization of the Report

This final report on the SBIR Phase-II EOS workstation project relies heavily on the documents that were generated during the development work and are included in the attachments. These documents are largely self-explanatory and describe in a coherent fashion the evolution of the software from functional requirements via the analysis to specific users' manuals for particularly mature components of the software.

Apart from these attached documents, we describe in nine separate chapters various aspects of the EOS workstation project and experiences. In the initial chapters we summarize the current status of remote sensing analysis software and the discrepancy between current sensing technologies and abilities of commercially off-the-shelf software systems.

The functional requirements are discussed in summary form in Chapter 3 with more details found in Attachment A. The issue of a target platform for EOS-type image analysis is discussed in Chapter 4 with some recommendations for future decision-making by EOS scientists. Chapter 5 presents the specification for the software which is presented in detail in Attachment B. A matter of grave concern is the availability of third-party software on a particular target platform. Issues surrounding this third-party software availability are being discussed in Chapter 6.

A major accomplishment of the effort is the creation of a so-called "digital light table" which permits a user to interact with very large overlapping images that can be viewed stereoscopically. In contrast to previous software this supports not only the function of a light table but also those of mono and stereo comparators maintaining a stereo cursor in a very unique manner that deviates from conventional ways in which stereo systems are being implemented.

A new concept that is largely inspired by EOS-type sensing is the "image cube." This is being separately discussed in Chapter 8 and is being illustrated there. A final set of

chapters describes the current status of the software, plans for a Phase-III and concludes with recommendations and an outlook for workstation considerations in the EOS-era.

CHAPTER 2

CURRENT STATUS OF EXISTING REMOTE SENSING ANALYSIS SOFTWARE

Remote sensing analysis software can be grouped into three categories:

- a. Commercial off-the-shelf (COTS) software that is supported by commercial endeavors and “lives” as hardware platforms change.
- b. Supported public domain software which has a general purpose and is typically not available on many platforms.
- c. Special purpose software that is applications-oriented, specific to this application and therefore limited in its functionality, again in the public domain.

2.1 General Purpose Commercial Software

Remote sensing software is in reality image processing software. The role of commercial off-the-shelf (COTS) software is very limited, although there has been a significant change since about 1989, about the time when the current project began. Up to that point, image processing software was large, bulky, expensive and only available on very specific operating systems and computers. This was dominated by special purpose hardware such as that offered by Gould De Anza, Dipix, I²S, Recognition Concepts, VICOM, Context Vision, GEMS and others. Hardware independent software had only very limited availability from small companies and was mostly distributed on PC platforms under the MS-DOS operating system.

Since 1989 the prevailing computer standards have begun to dominate in image processing and remote sensing software and a transition was experienced from the

— — — — —

various parochial operating systems to UNIX. There exist perhaps five to ten leading commercial vendors of remote sensing-type imaging processing software which all offer their product under the UNIX operating system and stake the claim that it can run on off-the-shelf RISC-type workstations or PCs.

This change has led to a standardization of the user interface using X-Windows and Motif widget sets, representing highly sophisticated graphical user interfaces (GUIs). In terms of functionality most image processing systems for remote sensing do not really *interact* with the image, but instead set up a "job" for batch-processing of images that exist on a permanent storage device such as a magnetic disk. Actual interaction with an image is, therefore, not typically supported even at this time. Representatives of COTS remote sensing analysis software that one can call "leaders" in the market include products by ERDAS, Terra-Mar, PCI and ER-Mapper. All of these organizations now support UNIX-based software that has high performance and very pleasing GUIs. None of them, however, supports the modern concept of interaction with multiple images on a screen using stereoscopy.

2.2 The Role of Image Processing Boards

Until 1989 high performance image processing was dependent upon specialized image processing computers or image processing board sets used in conjunction with high performance computers. Since 1989, when this project started, the newer RISC chips have become of such high performance that auxiliary boards are often no longer required. High performance RISC-type workstations of Hewlett-Packard/Apollo, Silicon Graphics Incorporated (SGI), IBM and SUN provide compute power that in many cases is sufficient and, therefore, does not need specific image processing boards. This, of course, has led to a decay in the segment of industry supporting specialized image processing hardware.

The only element of computing that still depends upon specialized hardware is the display for interaction with images. In the event where large images need to be

— — — — —

displayed in many colors very rapidly, some specific high performance real-time computing and visualization support may still be needed. However, it is expected that the next generation of 64-bit chips, such as the Alpha chip announced by Digital Equipment Corporation will even eliminate that need.

2.3 Commercial Data Visualization Software

An interesting development in the commercial off-the-shelf software for image processing is unrelated to remote sensing. From the aspect of general scientific data visualization, we have seen software packages such as IDL and PV-WAVE emerge to support general engineering work. These packages can then be used for remote sensing as well. One major limitation that such packages have is that they are typically not dealing with very large image data sets which are more typical of the remote sensing application. However, for applications that have only small images to deal with, say for example 1,000 x 1,000 pixels, these packages are perfectly applicable and offer a wide-range of capabilities. These capabilities, however, typically also exclude those that are needed for EOS remote sensing application, but can be added.

2.4 General Purpose Public Domain Software

There certainly exists stagnation in the public domain software for remote sensing image processing. The leading software packages in this area have been VICAR from NASA's Jet Propulsion Laboratory, ELAS from NASA's NSTL, and LAS from NASA's Goddard Space Flight Center. These packages were created at government research labs at the time where image processing was confined to large centralized computing facilities. These packages have typically not been ported into a flexible interactive workstation environment based on UNIX. Therefore, interest in these packages has been reducing and instead a multitude of COTS packages has come about to replace these general purpose public domain packages.

— — — — —

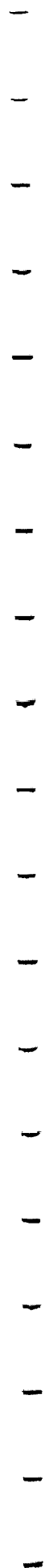
Various attempts have been made to get public domain software to migrate to the modern workstation world. These attempts have so far largely failed. For example, JPL's VICAR is still not available under UNIX and can therefore not compete with the flexible programs offered by the vendors mentioned under Section 2.1. It is also highly unlikely that it will make sense in the future to have these packages converted under government funding when low cost commercially available packages exist at this time.

An interesting development is the highly graphical software being produced under NASA's NRA funding at the Jet Propulsion Laboratory under the name of LINKWINDS, authored by A. Jacobsen at JPL. This software is competitive with that described earlier under the IDL and PV-WAVE name. Again one may argue that it is not meaningful to have public domain software written when some commercial capability is already available from existing sources.

2.5 Special Purpose Software for Remote Sensing Image Processing

There is a vast array of special purpose software that various research laboratories and centers of excellence have been creating to address oceanographic, meteorological, geological and other geoscience applications. The typical image processing component in those systems is fairly small and rather generic. What makes these packages unique is their ability to address a specific data format on the input side and provide representations of results that are of relevance to a particular applications domain. At times, also specific image analysis models are implemented to accomplish an applications-oriented result.

The most interesting recent developments under the special purpose software are developments relating to the analysis of "image cubes." Such software has been written at the U.S. Geological Survey in Flagstaff, Arizona and at the Center for the Study of the Earth From Space (CSES) in Boulder, Colorado. It is interesting to note that the specialized application software that was developed at CSES for the analysis of AVIRIS image cubes builds on top of the existing commercially supported software IDL. This is



a trend that is significant and points towards a path into the future. It generally offers a richer processing range than the in-house system of the USGS, and was written in a fraction of the time, simply by relying on successful commercial base software. It is to be noted, however, that generic software for image visualization remains to have its limitations with the size of data sets and with a current inability to support the display of multiple images in a stereoscopic mode.

2.6 Lessons Learned From A Review of Existing Remote Sensing Analysis Software

We can conclude from the analysis of the remote sensing software that existed at the beginning of this project that the most recent concepts for image processing were typically not available to the scientific workstation user. In the meantime the commercial market has caught up to some extent and has learned to capitalize on the availability of standard graphic user interfaces; also costs of software have come down dramatically. The issue now exists to build a useful system from existing COTS software rather than competing with such software by rewriting or creating from scratch.

The most important functionalities from an EOS point-of-view certainly were not implemented anywhere in 1989. However, to some extent some batch processing of modern EOS-type data sets has become feasible in a few selected cases. But even at this time no coherent and comprehensive capability exists to address the wide-range of difficult EOS-type remote sensing analysis tasks. This is why the current EOS workstation effort is of great significance. As we will show in the remainder of this document, we have tackled and solved a number of very important tasks that are significant for a successful interaction of the scientific user with EOS-type data sets.

—

CHAPTER 3

FUNCTIONAL REQUIREMENTS

3.1 Background

In this chapter we describe the top-level functional requirements of the EOS workstation prototype. A pictorial overview is shown in the corresponding top-level data flow diagram (Figure 5.1). A more detailed description of the functional requirements is provided in the EOS Functional Requirements document that was developed as part of the SBIR effort.

The EOS workstation should be able to efficiently process the future EOS images. We expect that these images will be from several different sensors with different resolutions as well as different spectral bands (SAR, HIRIS, MODIS). In addition, the workstation must process external data not acquired by EOS sensors.

In order to efficiently process EOS images, the workstation must allow the user to visualize the images and the features linked with them, relate images one to another, acquire features in one or several images, compute feature cartographic coordinates from image coordinates, and relate external data to images.

3.2 Overall Description

The EOS workstation is a standalone workstation that accepts images and ancillary data, DEM's, and ground features as input, and creates orthorectified images and features as output. It can be viewed as a tool box containing the necessary elements to perform complex processing, such as change detection and the use of ground truth for comparison or generalization.

3.3 Visualization Requirements

Visualizing images

- reduce the number of bands if it is greater than 3;
- scroll the image(s) on the screen if the size of the image is greater than the screen;
- display several subsets of bands one at a time if the number of bands is greater than 3;
- increase or decrease the the contrast and radiometric range of the displayed images;
- provide stereo capability, where appropriate, to improve the accuracy of point selection and add another dimension to the perception;
- zoom the displayed image to increase the resolution.

Generating perspective views

- register an image to a DEM;
- generate the perspective view.

Viewing images simultaneously with graphics

- display the features at the same resolution and in the same window as the displayed images.

3.4 Geometric Processing/Requirements

Image to ground registration

- find the corresponding location on the ground for each pixel in the image;
- acquire features known by their geographic coordinates;
- estimate the mapping between the image and the ground;
- resample the image.

Image-to-image registration

- acquire common features in the two images;
- estimate the mapping between the two images;

- resample one image.

Referencing features in an image

- model the mapping between the feature reference and the image reference.

Computing feature cartographic coordinates

- if the feature is identified only on one image, then apply the image to ground grid;
- if the feature is identified in two images, then compute the stereo intersection.

3.5 Target Platform Constraints

The main constraints are:

- software
 - operating system, UNIX;
 - programming language, C;
 - user interface built on the X-Windows system.
- hardware
 - Ardent Titan II computer with:
 - * 32 Mb of memory,
 - * 2 GB of disk storage,
 - * 24 bit image display (G3 controller),
 - * keyboard,
 - * mouse.
 - Tektronix liquid crystal plate (for stereo display).

The basis for these target platform constraints is described in Chapter 4.

—

CHAPTER 4

SELECTION OF A TARGET PLATFORM FOR AN EOS-TYPE WORKSTATION

4.1 The Original Concept From the SBIR Phase-II Proposal

It was originally recommended that the computer platform for the EOS-type work would be a RISC-type workstation manufactured by SUN with a specialized image processing board manufactured by VITEC. The reason for this configuration was that at the time of the proposal, in 1989, the concept of a specialized image processing board to accelerate a UNIX-based workstation was the only means by which one could accomplish stereo viewing of large black and white and color images.

The real concern at the time of the proposal was the question whether high performance computing was feasible on a single-user workstation and whether high performance computing was not by virtue of its cost relegated to departmental computing systems that would serve multiple users simultaneously. A parallel effort at VEXCEL was the development of the so-called Geophysical Processing System (GPS) to process the expected images from the European Remote Sensing satellite ERS-1 in such a manner that certain routine products would be obtained from overlapping sea-ice images, namely motion maps, classification maps and ocean wave spectra.

This type of "departmental" or applications-specific computing would serve many users and would not really be interactive at a users desk.

Upon reflection the alternative to this departmental computer concept as originally discussed in the proposal is contrasted with the desktop workstation in the form a personal computer. At the time of the proposal, these were computers with 16-bit CPUs in the form of PC 286's. As work for the EOS workstation began, the first 386 PC's became available and we are currently seeing the use of 486 PC's and are awaiting

introduction of the 586 and 686 PC's. However, the difficulty with the personal computer remains its relative slow bus structure with the AT bus. For high volume data sets that need to be moved very quickly from permanent storage to an interactive display, this is a limitation that was to be avoided.

4.2 Selecting A Graphics Super Workstation Computer as the Target Platform for an EOS-Type Workstation

In between the concept of a departmental computer and the concept of a personal computer emerged the concept of the graphics super workstation. Initially there were two vendors of such workstations, namely Silicon Graphics Incorporated (SGI) and Stardent Computers (nee Stellar and Ardent). Upon close inspection it was revealed that for applications to pixel-based natural images, Stardent Computers offered the more flexible and powerful product. In contrast, SGI offered a product that was specifically effective with graphics data and computer-generated images as opposed to natural images.

Upon an intensive exchange with the program's COTR and in light of the expected change in computer hardware costs, it was decided that the proper target platform should not be a personal computer nor should it be a departmental computing system but it should be a workstation of the category designated as graphics supercomputer.

The budget that was originally proposed for the departmental computing system was also covering the choice of a graphics supercomputer. While it was clear that an individual scientist would not immediately be able to afford a personal workstation for a hardware cost in excess of \$100K, it was expected that costs would decline over the lifetime of an SBIR Phase-II and a subsequent SBIR Phase-III project so that by the time a product would be introduced into the market, hardware costs would have become affordable for individual scientists.



This overall consideration is currently proving valid. The performance of graphics supercomputers that was available for costs in excess of \$100K in 1989 is expected to be available by 1993 for a cost of \$20K. One therefore reaches the budget that one may typically expect to be available for high performance capabilities at the desk of the individual scientist. The overall driver towards a graphic supercomputer was the avoidance of specialized third-party image processing boards that would instantly cause the workstation computer to become bulky due to specialized and machine-dependent software needs. With the advent of faster chips, in particular currently 64-bit RISC chips, specialized image processing boards are expected to no longer be necessary for high performance. Instead a graphics supercomputer, such as models to be developed from new 64-bit chips (or models currently being introduced by Hewlett-Packard/Apollo, DEC and others) all satisfy the computing requirements that one may have at a deskside to cope with the massive amount of EOS data.

4.3 The Choice of the Target Platform

The above considerations led to the conclusion that instead of a SUN workstation computer and a VITEC image processing board, a Stardent 3000 graphics supercomputer should be selected which supports stereoscopic viewing under X-Windows, has high performance floating point operations capabilities and numerous native data visualization tools such as the Doré software, the application visualization software AVS and a native implementation of MATLAB.

4.4 Experiences With the Choice of Target Platform

At the start of this project in 1989, it certainly appeared as if the particular computer vendor, Stardent, might acquire a significant portion of the graphics supercomputer market. This led to the expectation that necessary third-party software packages would rapidly be ported onto this computer and would, therefore, be available for integration by the project.

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

This expectation did not materialize. The computer system, which competed strongly with Silicon Graphics IRIS workstations, was unable to attract the attention of software vendors and as a result expected software ports did not take place. This was particularly harmful for this project because important software components, while promised by the vendor, never became available during the life of the project. Missing were in particular:

- a relational data base software package (for example INGRIS);
- a Geographic Information System package (for example ARC-INFO);
- a generic commercially-support image processing software package (for example IDL, PV-WAVE or ERDAS).

These shortcomings represented a handicap for the project because a considerable amount of time had to be expended by project personnel to overcome the absence of these third-party components. However, since Phase-II is only a prototype implementation of certain software concepts and not a deliverable product, these shortcomings by themselves are not detrimental. In the transition to a Phase-III a fresh look can be taken at the target platform and that will have to incorporate thorough consideration for the availability of third-party software.



CHAPTER 5

EOS WORKSTATION SPECIFICATIONS

This chapter describes the specifications for the workstation. Each subsection explains the numbered bubble in Figure 5.1.

5.1 Detailed Requirements

5.1.1 Visualize Images

The image visualization function is the root of this workstation. Most of the user interaction with the system will be done through the visualization of images and features.

Input

- image(s),
- viewing parameters.

Output

- displayed image windows.

Processing

- animation of images bands through the definition of a window and display frequency;
- scroll;
- zoom;
- radiometric processing;
- band selection.

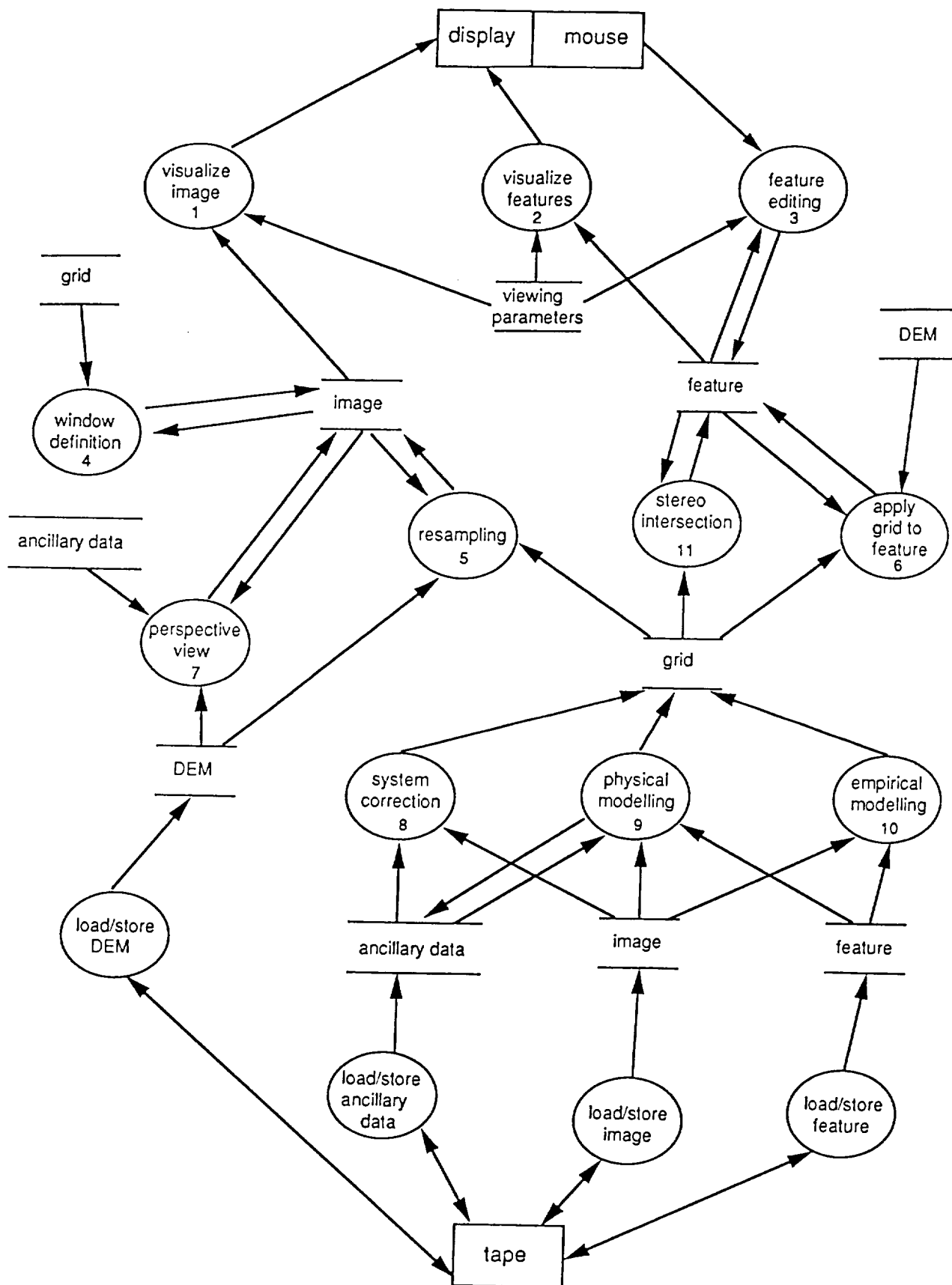


Figure 5.1: EOS Workstation data flow diagram.

5.1.2 Visualize Features

The feature visualization function also uses the image visualization. Features will always be visualized with images.

Input

- features referenced in the image,
- viewing parameters.

Output

- features to display.

Processing

- select the features totally or partially included in the images windows,
- transform feature absolute coordinates to window coordinates,
- apply zoom factor to feature window coordinates.

5.1.3 Feature Editing

Feature editing is mainly used for two purposes:

- Thematic Study: In order to study the spatial and radiometric properties of a particular item, one needs to define it spatially in one or two images.
- Geometric Processing: One must acquire common features between one image and the ground or between two images in order to model the mapping between them.

Input

- feature list, these are features already acquired in one image and the ground or in two images;
- viewing parameters;
- cursor coordinates.

Output

- updated feature list.

Processing

- visualize images and features;
- interactively modify the graphic display and thereby update the feature list.

The update options are delete feature, modify feature and acquire feature.

When possible a correlation will be used, between two images, to refine the feature coordinates.

5.1.4 Window Definition

An image window is useful for emphasizing the area of interest, and reducing the processing time.

Input

- image,
- grid.

Output

- window.

Processing

- if the window is defined in image coordinates, extract the window defined by the user;
- if the window is defined in cartographic coordinates:
 - * if the image is geometrically corrected, use the image parameters to translate the cartographic coordinates into image coordinates;
 - * if the image is geometrically raw, use the grid (ground --> image) to translate cartographic coordinates into image coordinates;
- real-time.

5.1.5 Resampling

This function resamples an image in order to register this image with another reference.

Input

- image,
- DEM,
- grid.

Output

- resampled image.

Processing

- if there is a DEM:
 - * interpolate the grid to the DEM grid steps;
 - * modify the image coordinates at each node with the DEM and (dLine/dHeight,dPixel/dHeight) values;
- generate the output image.

The user will specify the method of resampling (nearest neighbor, bilinear, bicubic) and the number and names of the bands to resample.

5.1.6 Apply Grid to Features

This function transforms the feature coordinates expressed in one reference to another reference. This function is useful for the following:

- view on one image the location of external features (referenced only geographically);
- look, in one image, the features acquired in another image without explicitly acquiring the features in both images. This can be useful for change detection or radiometric study.

Input

- feature list: expressed in one reference (image or ground);
- grid: mapping the transformation from the feature list reference to the new reference;
- DEM.

Output

- feature list expressed in the new reference.

Processing

- for features acquired in a geometrically corrected image to be references in the raw image:
 - * using the image coordinates of each point of the feature, compute their cartographic coordinates;
 - * find the nodes of the grid (ground --> image) surrounding the feature point;
 - * interpolate the raw image coordinates;
- for features acquired in the raw image to be referenced on the ground (with DEM):
 - * find the surrounding grid nodes (image --> ground) of each point of the feature using the raw image coordinates;
 - * interpolate the corresponding cartographic coordinates;
 - * using the grid (ground --> image) viewing direction slope, compute the cartographic coordinates of the intersection of the DEM with the viewing direction;
- for features acquired on the ground to be referenced in a geometrically corrected image (without DEM):
 - * using the cartographic coordinates of each point of the feature, find the grid surrounding nodes (ground --> image);
 - * using the grid (ground --> image) viewing direction slope, iteratively project the feature cartographic coordinates at the grid reference height;
 - * using the grid (ground --> image) parameters compute the image coordinates.

5.1.7 Perspective View

The perspective view shows on one image the radiometric content plus another dimension.

Input

- image,
- DEM,
- platform data.

Output

- perspective view.

Processing

- define the perspective view geometry,
- define the common area between the image and the DEM,
- generate the perspective view,
- the number and names of the bands to resample.

5.1.8 System Correction

The System Correction function computes the projection of a raw image onto the ground. It assumes the existence of ancillary data associated with the image.

Input

- image parameters,
- ancillary data,
- platform data errors.

Output

- one grid describing the mapping from the raw image to the ground;
- one grid describing the mapping from the ground to the raw image.

Processing

- define a regular grid in the raw image;

- for each node in the image grid:
 - * compute the viewing direction in the sensor reference,
 - * compute the viewing direction in the platform reference,
 - * compute the viewing direction in the earth reference,
 - * intersect the viewing direction with the earth model;
- invert the above grid.

5.1.9 Physical Modeling

This function estimates errors in the platform data using deviations between the computed and acquired cartographic coordinates of the features. This is possible only if the ground features are referenced in the geometrically raw image.

In the case of two geometrically raw images, there are two different scenarios:

- If a DEM is available, then estimation of the platform data errors for both images is possible.
- If a DEM is not available, then estimation of the platform data errors of only one image is possible.

Input

- image parameters;
- features acquired in the two images or in the image and the ground;
- ancillary data;
- DEM.

Output

- one grid describing the mapping from image 1 to the ground (or image 2);
- one grid describing the mapping from the ground (or image 1) to Image 1;
- platform data errors.

Processing

- select the features to be used in the modeling;



- estimate the coefficient of the function representing the error in the platform data;
- compute the residuals for active and control features;
- compute the grid from the image to the ground;
- invert the previous grid.

5.1.10 Empirical Modeling

This function estimates the mapping between two images or between one image and the ground using the ancillary data.

Input

- image parameters,
- features common to the two images, or to the image and the ground.

Output

- one grid describing the mapping from the image to the ground (or other image);
- one grid describing the mapping from the ground (or other image) to the image.

Processing

- select the features to be used in the modeling;
- estimate the coefficients of the function mapping the deformation between the two images. The functions are: polynomial with different degrees and “rubber sheet” algorithm.
- compute the residuals for active and control features;
- generate the two grids.

5.1.11 Stereo Intersection

This function computes the cartographic coordinates of a feature acquired in two images by stereo intersection.



Input

- feature,
- grids image 1 <--> ground,
- grids image 2 <--> ground.

Output

- feature.

Processing

- for each image compute the cartographic coordinates by interpolating the image to ground grid, and compute the viewing direction vector by interpolating in the ground to image grid;
- intersect the two viewing directions.

5.1.12 Load/Store

This set of functions is the main interface with the external world. The main data that we will load or store are DEM's, ancillary data, images and features.

Input

- object to store or load.

Output

- object to store or load.

Processing

- specific to the type of data.

5.2 Usage Scenarios

Up to this point the EOS workstation has been described as a tool box. In this section we describe some of the “macro functions” that can be performed using these tools.

5.2.1 Register One Image to Another Image

This function registers one image to another image.

Input

- image 1,
- image 2.

Output

- resampled image: image 1 or image 2.

Processing

- define the window in each image containing their intersection (window definition);
- put the two images at the same resolution (standalone zoom). This will be useful for stereo display (if required);
- edit features in the two images (feature editing);
- modelize the correspondence between the two images (physical or empirical modeling);
- resample one image (resampling).

5.2.2 Ortho Image Generation

This function generates an image registered to the ground.

Input

- image,
- DEM.

Output

- ortho image.

Processing

- acquire common features between the input raw image and the ground (feature editing);

- estimate the platform data errors (physical modeling);
- generate the grid from ground to raw image (system correction);
- resample the image using a DEM (resampling).

5.2.3 Project External Features in Image

This function will generate the image coordinates of features referenced on the ground.

Input

- feature list,
- image.

Output

- feature list.

Processing

- acquire common feature between the image and the ground (feature editing);
- estimate the platform data errors between the image and the ground (physical modeling);
- compute the grid from the ground to the image (system correction);
- compute the image coordinates of the feature list (apply grid to features);
- visualize the result (visualize images and features).

5.2.4 Change Detection

Change detection involves external features acquired at a particular date and an image acquired at a later date (mainly the same as project external features in image).

Input

- external feature list,
- image.

Output

- feature list: same feature as input but acquired on the image.

Processing

- acquire common feature between the image and the ground (feature editing);
- model the correspondence between the image and the ground (physical or empirical modeling);
- compute the image coordinates of the feature list (apply grid to features);
- edit the external feature with the image and acquire the same features from the image (feature editing);
- compare the two sets of features.

5.2.5 Mosaicking

Mosaicking is the union of images with the same reference system. Here we assume that there is no geometric distortion between the images (if any, we must first model the deformation between each pair of images).

Input

- ortho images.

Output

- mosaic.

Processing

- If there is discrepancies between adjacent images:
 - * acquire common points onto the images (feature editing);
 - * model the deformation locally (the deformation function must be defined over the entire image and be the identity outside the intersection of the two images);
 - * resample each image (resampling);
- define the boundaries of the useful part of each image (feature editing);
- adjust the radiometry at the images edges.

CHAPTER 6

ISSUES CONCERNING THIRD-PARTY SOFTWARE

6.1 Generic Image Processing Software Package

The overall concept of the EOS workstation as illustrated in Figure 1.1 must rely heavily on the availability of existing software, either of a generic nature or applications specific. As we discussed in Chapter 2 about the current status of remote sensing analysis software, one needs to expect that the EOS workstation operates with a third-party generic image processing software package. Numerous software packages were investigated for usefulness in the EOS workstation. In particular the project investigated the usefulness of ELAS, LAS, VICAR, PCI's EASY-PACE and ERDAS. It was quickly revealed that the desired functionalities did not exist in the public domain because the software was either not really available in a robust manner in UNIX or it didn't have the compatibility with an X-Windows graphical user interface or it completely ignored the need to interact with images.

A sincere attempt was made to investigate the usefulness of VICAR as a very well-known software package that was offered at the Jet Propulsion Laboratory. Unfortunately, great effort has already been expended by various groups throughout the United States to port VICAR into a UNIX operating system using X-Windows as the user interface. None of these efforts has really been completed and VICAR remains largely unavailable in this environment.

The desirable solution would be to employ a third-party software package by one of the commercially active vendors. Unfortunately, the preferred system, namely IDL by RS, Inc. or its derivative, PV-WAVE of Precision Visuals, Inc. were not ported to the target platform during the lifetime of this project.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

However, we can report that in the transition to Phase-III of the SBIR the conclusion now is to employ IDL by RS, Inc. and integrate it into the EOS software.

6.2 Geographic Information System Software

The administration of graphical map data, consisting of points, arcs, polygons and alpha-numeric data, is an important element of working with remote sensing studies. It was expected at the beginning of the project that public domain software, in particular a raster-based system such as GRASS, would be integrated into the EOS workstation. As a result a copy of a UNIX version of GRASS was obtained and was installed on a SUN computer. The expectation was that a conversion of that software to the target platform could be accomplished. This hope proved futile. The resources to port GRASS to the Stardent 3000 proved beyond the scope of the SBIR Phase-II project and was, therefore, abandoned. However, many of the capabilities of GRASS were studied and upon completion of the SBIR Phase-II project, it now appears that an X-Windows-based UNIX implementation of GRASS that operates in a robust manner seems now to be available when it was not available during the lifetime of the Phase-II project. It is expected, therefore, that for a Phase-III implementation the work done under Phase-II with GRASS will still bear fruit. The work done with GRASS is described in a separate Attachment E.

The most desirable GIS software package would be ARC-INFO from ESRI in Redlands, California. The desirability of this software is a result of its wide availability as a quasi-standard in the industry. We came to the conclusion that this software could be available in one of two forms:

- a. As a stand alone capability hosted on a PC connected to the EOS workstation via a local area network and some import and export routines that permit one to extract work files from the GIS and import the modified work files back into the GIS. This kind of approach has been used in a different project by VEXCEL where a GIS was combined with limited image processing for the application to

a system for collecting municipal planning data from aerial photography (Williams, 1992).

- b. The availability of ARC-INFO on the EOS workstation itself. This implementation was hoped to be available during the lifetime of the project, but the vendor of the software, ESRI, Inc., did not port the software to this platform.

As a result of the experiments with the GIS software, we are now in a transition to Phase-III very concerned about the availability of a quasi-standard GIS software package such as ARC-INFO. Phase-III will, therefore, incorporate that software on the platform itself.

6.3 Relational Data Base Software

The massive amounts of images and meta data about images require that a considerable amount of planning and record keeping is implemented on the workstation. Clearly, this was expected to be done with a commercial off-the-shelf relational data base software. It was specifically recommended in the proposal that the workstation be built with the help of INGRES. The computer vendor guaranteed that that software would be available any time after delivery of the computer. However, during the lifetime of the project INGRES had never been made available on the target platform. Therefore, many of the functions in the relational data base program had to be performed in a generic fashion on the computer and absorbed more development effort than was originally planned.

Again, in the transition to Phase-III a very careful plan will consider that a major COTS relational data base software package, such as INGRES, will be available and will be the core of administrating meta data and other relevant information. The GIS itself, in particular ARC-INFO, is built itself around a relational data base system, namely INFO. One may consider to rely on the relational data base that supports the GIS rather than having a stand alone relational data base system implemented. This question could not



be meaningfully resolved during the Phase-II project because neither ARC-INFO nor INGRES ever became available to the project. However, resolution of this issue must occur in the transition to Phase-III.

6.4 Application Specific Third-Party Software

It was the intent to configure the EOS workstation in such a way that various application specific software packages can be hooked into the it. This was to be demonstrated by a small number of such packages. As a part of the EOS workstation effort, we decided that a digital elevation model manipulation package had to be made available. In this manner images and topographic data can be resampled and visualizations of image and topography can be supported. We selected a digital elevation model package manufactured by Radian and denoted as CPS-3. This package originally did not run on the target computer and was, therefore, ported by the project onto the target computer as a third-party package.

An additional application specific package was for processing SPOT images and relating SPOT images to a world coordinate system. A third-party package was selected; it is being distributed under the name SPOTCHECK and was ported onto the target platform.

Furthermore, a prototype software package denoted as SPECTRAN was developed and integrated into the system. SPECTRAN was intended to serve as a tool for exploring spectral image cubes such as those produced from AIS, AVIRIS and HIRIS. The emphasis was placed on mineral identification and comparison with images from different sensors and other derived images. The functionality implemented was a subset of ISIS with the additional capability of having a reference cube of nonspectral data (see Appendix E). Currently the SIPS (Spectral Image Processing System) developed at CSES would be the ideal candidate for hosting on the workstation.

6.5 Visualization Software

Doré and AVS are two visualization software packages that came bundled with the target computer and represent a very attractive component of that computer. One may be swayed to use this particular computer workstation simply because of that software.

Doré serves the presentation of surfaces. The Doré Library provides functions for describing a scene and subsequently drawing it onto a display device. It serves both as a scene description data base and a visualization toolkit. Doré functions can be used to describe objects that compose a scene, define their characteristics, specify the viewpoint and the scene lighting when the scene description has been completed. Doré can display the objects in various ways, including shaded perspective renderings, wireframe drawings, and “point clouds.”

The Application Visualization Systems (AVS) is a new concept that revolutionizes the work with images on computers. AVS is a so-called “visual computing software” system which allows one to configure a sequence of image processing steps not by means of command lines or writing a program, but instead by fully employing a graphical user interface and composing a sequence of image processing steps very much like a composition of a puzzle. Figure 6.1 is an example of an AVS-type processing flow. This scenario produces surfaces that represent a match between AVIRIS spectral data and a prototype laboratory minerals’ signature.

- Input an image cube of spectral differences;
- subsample the cube;
- convert the data to floating point;
- generate an isosurface of the cube;
- render the surface and display the results.

1000

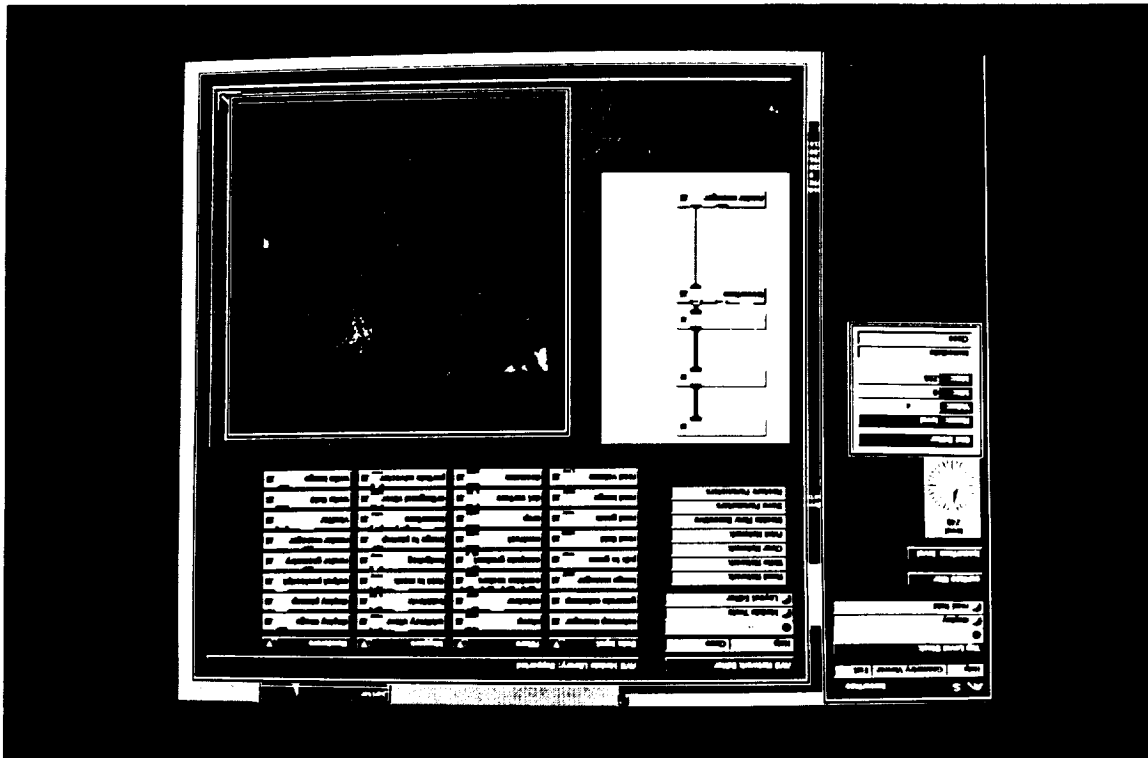


Figure 6.1: Example of an Application Visualization System (AVS) program flow. Note that the X-Windows-based user interface shows the data processing flow as individual boxes and connecting arrows. Color can be used to code the boxes representing different functional steps. This is denoted as “visual computing environment.”

CHAPTER 7

INTERACTING WITH LARGE OVERLAPPING STEREO IMAGES: THE DIGITAL LIGHT TABLE LT

7.1 Background

A major obstacle and limitation of existing image processing technology is the inability to deal with very large images in an interactive mode. This is completely unacceptable given the advent of graphical user interfaces, RISC-type processing workstations and fast buses connecting external disks to large refresh memories.

This was recognized early on in the field of image simulation. As a result PIXAR, Inc. developed what it called the "electronic light table." This was a tool to interact with very large individual images and allowed one to instantly roam, zoom, warp an image and also relate one image to another. No limitations did exist regarding the size of those images. However, this software was only available on very specialized hardware, available from PIXAR, Inc.

It was recognized early in the EOS workstation development that a tool is needed for the scientists to be able to bring up two color images and displace them with respect to one another and display them at arbitrary resolutions and be able to continuously roam through this image pair.

7.2 What is "LT?"

Therefore, a design was developed for the softcopy light table "LT." LT incorporates innovative software that provides the flexibility of a softcopy format in an environment dominated by hardcopy equipment. The software displays large digital images on a



single imaging monitor, in either stereo or monoscopic form, and digitizes the features that appear on the images.

LT's visualization tool displays and manipulates large-format mono or stereo images in both gray scale and color. It pans in the X- and Y-directions and zooms on the displayed images. In addition, its interface controls the brightness and contrast of the image, and modifies its horizontal and vertical image parallax to correct for image distortion. It also determines a reference point for accurate measuring.

LT superimposes related graphics data over the image and roams and zooms on both images simultaneously. For example, vector information is overlaid on the image to include data such as elevation contours, road and river networks, and geographical boundaries. The graphics data are perfectly tied to the underlying image so that roaming and zooming on the information is without losing the exact registration of the two images.

LT also supports digitization and acquisition of data points from the displayed images. One can specify single points, grids of points, breaklines, drainlines and tie points -- points that are common to three or more images -- on the displayed image. Because the image is displayed on-screen, one can revisit marked areas at any time and add, delete or append additional points. In addition, because one can see the image in stereo, one has a 3D representation of exactly where a data point on the image is placed. This ensures better accuracy and lowers or eliminates the amount of time one needs to spend editing the contour map created from the data points. Another feature LT offers is the ability to save permanent records of the points digitized on an image. One can save the images with the acquired data points intact and refer to it at a later date.

A unique software option of LT can create a height representation of the acquired data points in a separate window. This feature, the

Terrain Window, verifies the accuracy of the data points by allowing a visual comparison of the model one creates with the on-screen image.

The software technology that visualizes and digitizes images is identical for both the mono and stereo programs. The only difference between the two display systems lies in the hardware on which the image is displayed. One can display mono images on any machine running UNIX and X-Windows. However, to see stereo images, one must run LT on a monitor that supports polarization technology and wear a pair of polarized glasses. The glasses allow one to see one image with each eye to accomplish true stereo.

LT's menu-driven interface uses Motif and X-Windows to enable even novice users to run the program expertly with minimum training. The interface allows the user to make selections from a menu, and either use a standard knob box or an on-screen version of the box to change the cosmetic look of the displayed image.

LT's stereo capabilities provide the opportunity to compare images in real-time. The stereo and monoscopic display programs offer interactive roaming and zooming. In addition, the program's innovative digitizing tool allows one to acquire data points and lines more accurately than conventional systems. Better data collection provides more accurate input for the later creation of precise height models.

A complete manual for the LT software is enclosed as Attachment C. The experiences with this software have shown that it has become an invaluable and very important tool; it may be the most significant result of this effort. Since funding under the SBIR Phase-II did not allow to develop LT completely under this program, VEXCEL Corporation had to add separate and additional funds to complete LT. As a result, it is available as part of the EOS workstation but residual rights to that software remain with VEXCEL Corporation; LT is, therefore, not a public domain piece of software.



CHAPTER 8

CREATION OF, AND INTERACTION WITH, IMAGE CUBES

8.1 Creation of Image Cubes

The EOS suite of sensors is very much oriented towards the creation of multiple images of a study area of interest. Multiplicity of those images comes from an individual sensor which produces many component images such as spectral bands or polarizations; or having several sensors on one platform imaging the object into multiple records; or images are taken by several passes over an object to study the reflectivity of the object as a function of incidence angle or as a function of environmental conditions.

Customarily a multiplicity of images might be denoted as "image cube." Typically, an image cube is expected to consist of the individual spectral bands that are obtained by a single sensor event. A more complex problem exists if an image cube is to be created by individual images taken at different times from different sensors at different resolutions, or even from non-image data such as digital elevation models, drainage maps, vegetation boundary maps, etc.

A considerable effort was expended in the creation of the EOS workstation concepts to address the issue of co-registration of dissimilar images, for example from radar LANDSAT, SPOT or digitized National High Altitude Photography (NHAP). The problem domain of creating an image cube from dissimilar images goes far beyond the SBIR Phase-II funding. Therefore, the concept of creation of an image cube was merely illustrated by prototype software, but remains very preliminary. Figure 8.1 presents several elements of an image cube consisting of SPOT, LANDSAT, radar images and of USGS topographic data. Figure 8.2 illustrates the appearance of an image cube on the EOS workstation in an axonometric view. Once the image cube has been created, one can present to the user various intersections of the cube with planes. The difficulty in this element of the EOS workstation is not the presentation of the data

1

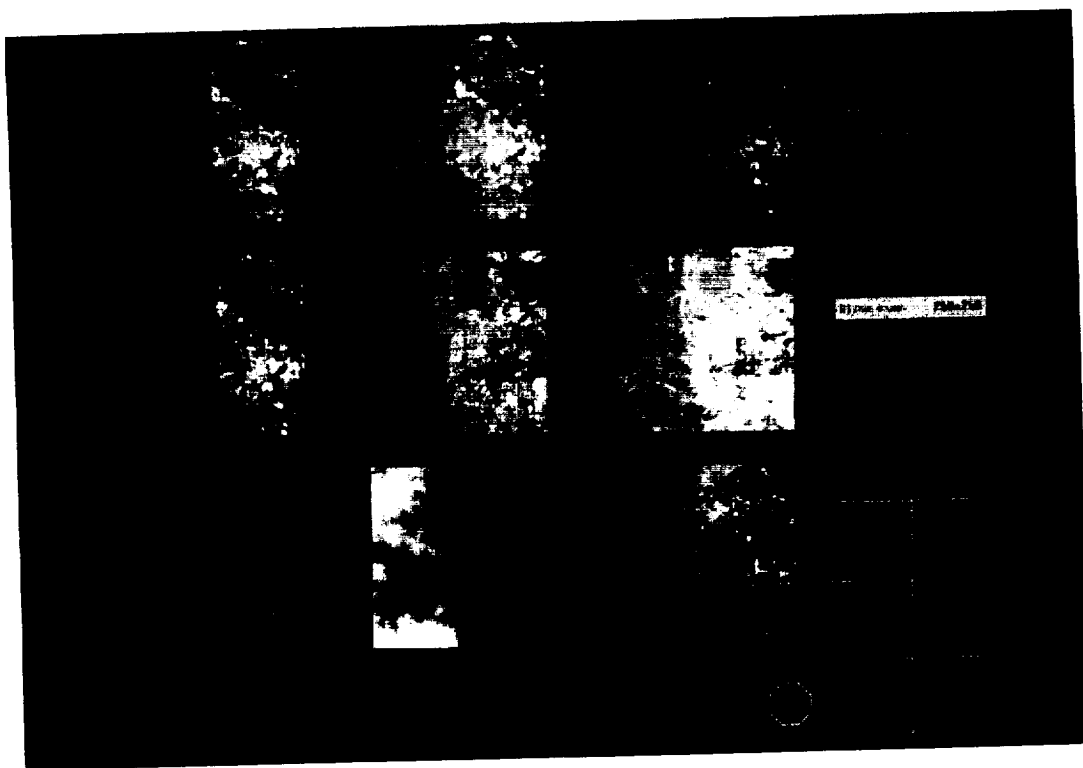


Figure 8.1: Components of a typical EOS-workstation "cube," consisting of co-registered images and non-image GIS data.

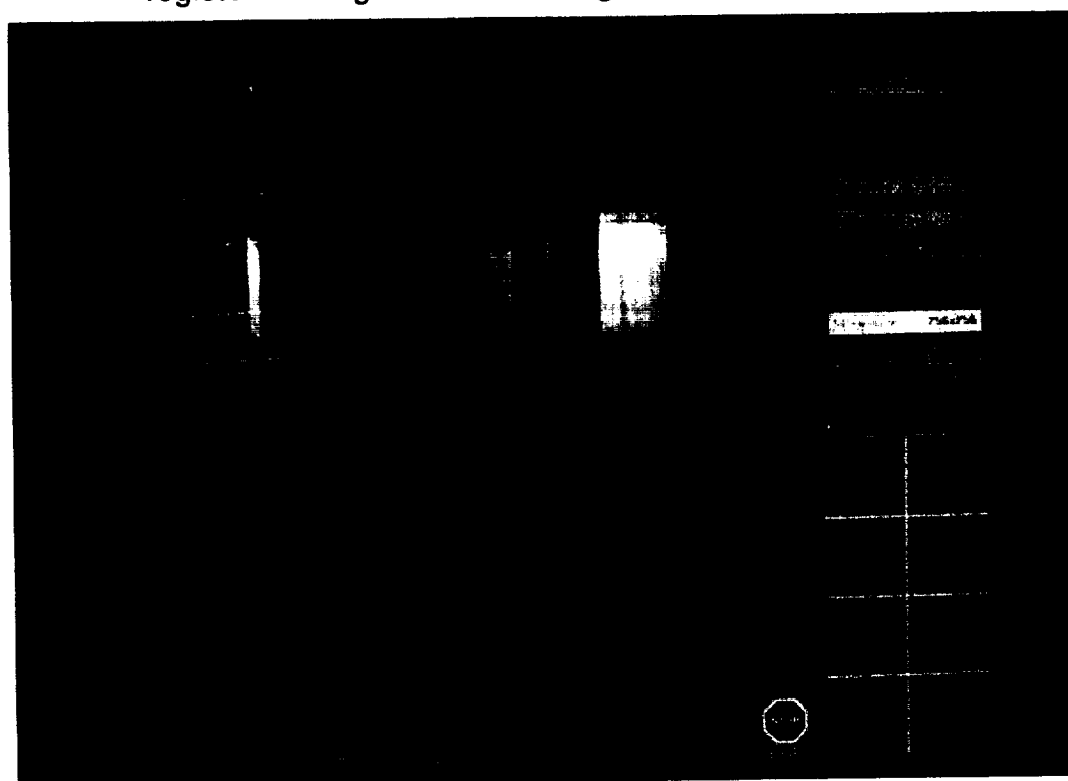


Figure 8.2: Axonometric view of an image cube with presentation of a "slice" through the cube.



because this can very largely rely on native visualization software that is part of any graphic super workstation; instead, the difficulty is in the creation of the image cube. Note that images from different sensors with different sensor models need to be co-registered into a coordinate system of the object. Therefore, individual sensor images need to be orthorectified using digital elevation data if they exist and subsequently these orthorectified images need to be precision co-registered if they don't match initially.

Figure 8.3 illustrates the super position of a color SPOT image and contour lines that are available from the U.S. Geological Survey's DEM-DLG data file. These contour lines must be projected from the object space into the image if the image has not been orthorectified.

8.2 Comparing Two Images

Upon examination of the needs of the scientific user, and once an image cube exists, it was learned that a major issue is the comparison of two images describing a phenomenon before and after a certain change occurred or relating images from different sensors to one another. A traditional approach for relating images from different sensors has been to create some combination using intensity hue saturation and the transformation into red-green-blue. Therefore, two or three images were presented as intensity hue saturation of a color image and then converted to red-green-blue for a false color representation. It has often been found that such presentations are very confusing to the user who would like to understand which contribution in an image comes from which sensor. Therefore, the EOS workstation adds an additional capability for image comparison that is novel in its concept and implementation.

Figure 8.4 illustrates how an Image 1 serves as a background and Image 2 is embedded on top of that background. Various ways exist now to present to the user both images at the same time as follows:

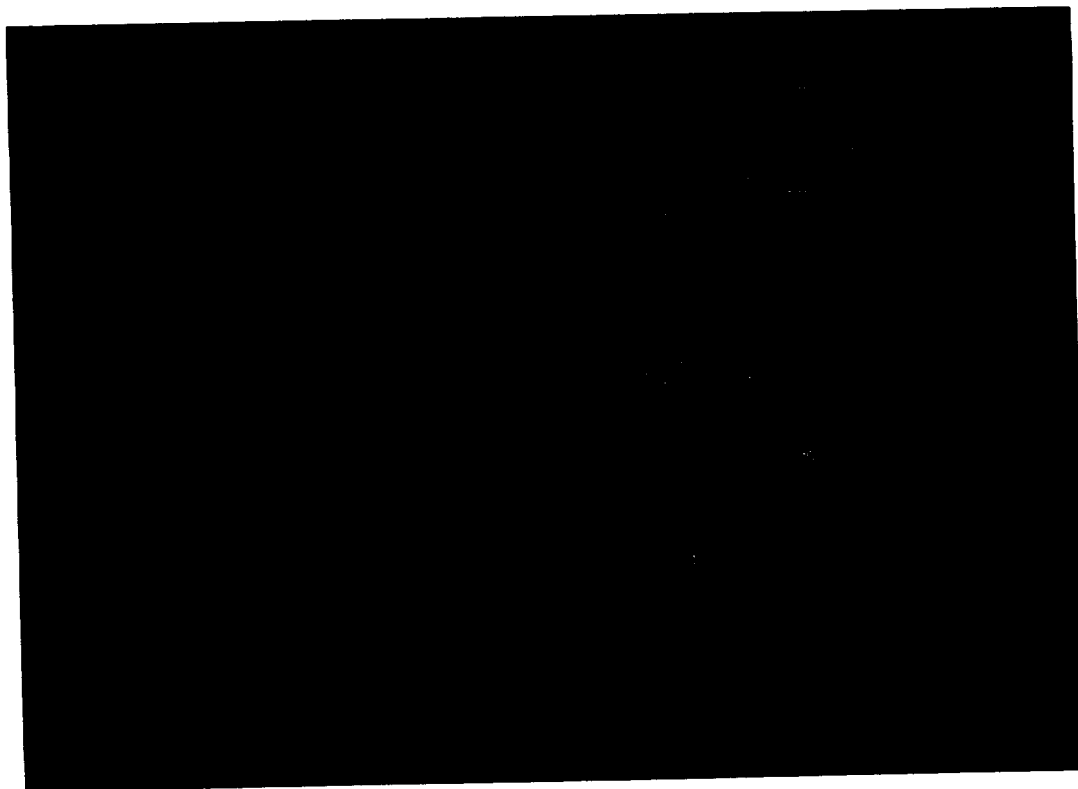


Figure 8.3: A special case of an image cube presentation of a SPOT image and elevation contours, which are, in this case, a raster image as well.

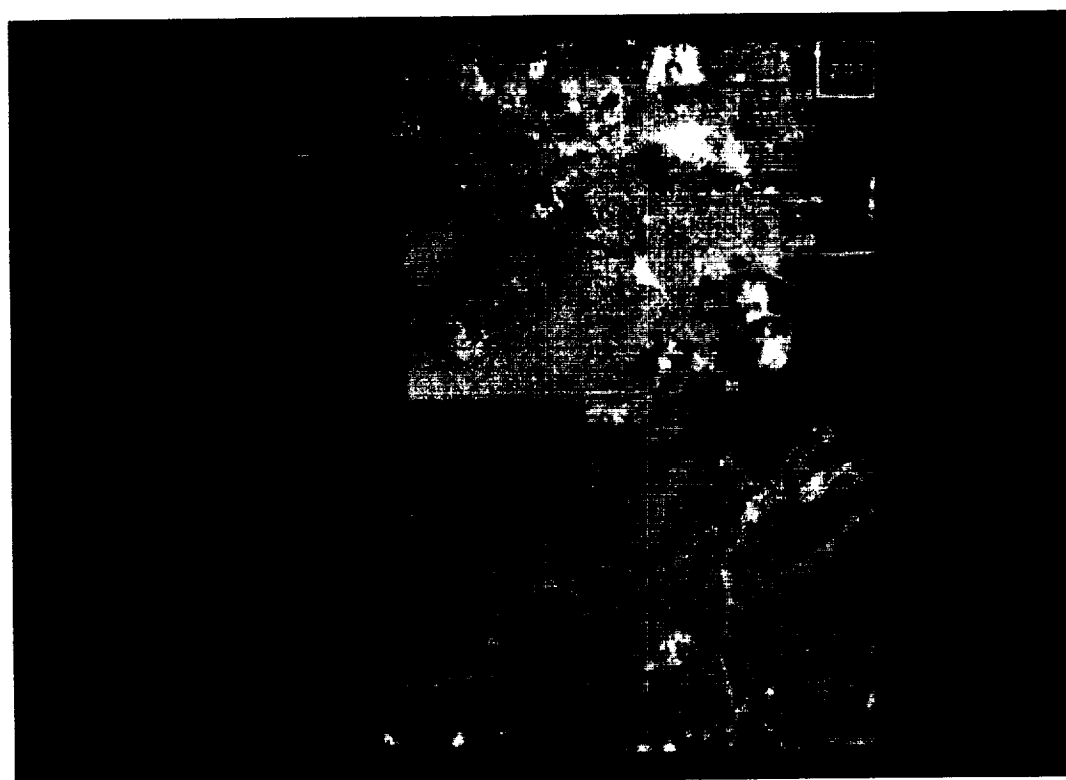


Figure 8.4: Embedding a local image window into a background image. Location of the windows can be "dragged" with the mouse.

— — — — —

- a. Image 1 completely covers up Image 2;
- b. Image 1 is translucent and one can see Image 2 and Image 1;
- c. Image 1 and Image 2 can flicker upon a push on the mouse button.

The important feature of this image comparison is the ability to move the smaller window over the background image. In this manner an instant comparison is feasible between the two images. A feature that appears of interest to the human observer in the background image can instantly be compared with its manifestation in another image. Also the image in the small window can be changed and scaled so that the background image can be at one scale and the small window is at a second scale. This capability is illustrated in Figure 8.5.

It may also be of interest to have several windows simultaneously displayed on the graphical user interface. Figure 8.6 explains.

And finally an application exists of comparing two color images with one another. While the comparison of black and white images has been implemented in such a manner that it is instantaneous, the same has not been feasible for color images. The data quantities required are significantly larger and a comparison of two color images, while feasible, is currently not instantaneous.

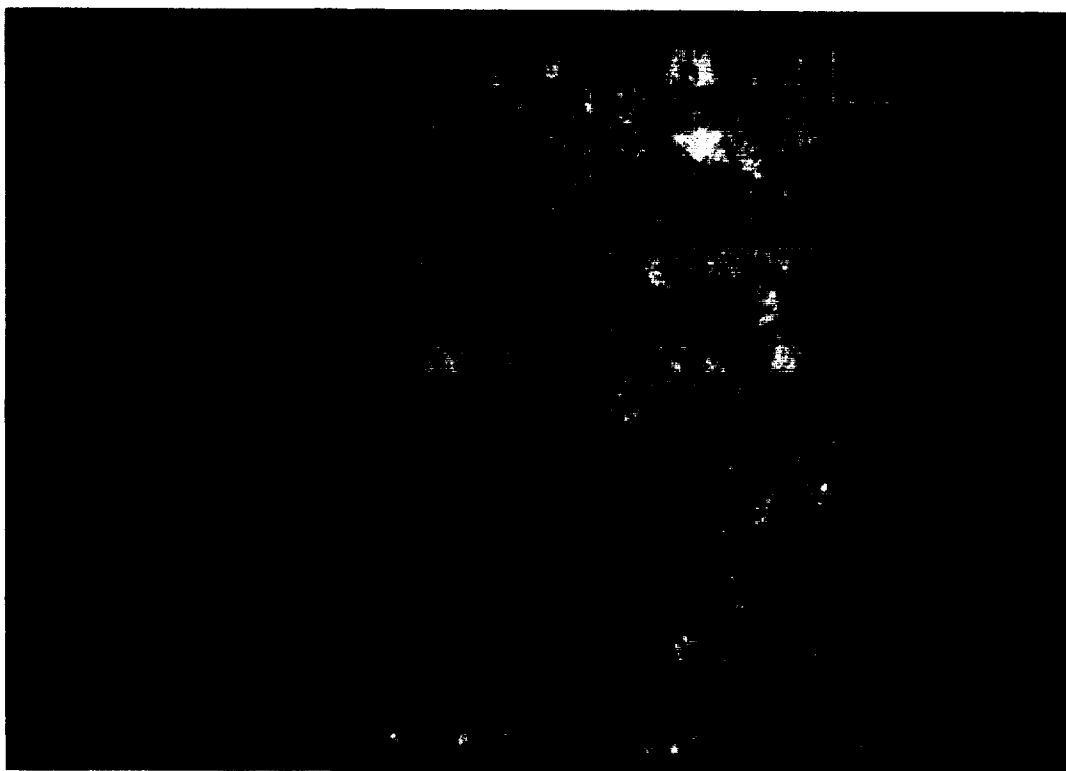


Figure 8.5: Same as Figure 8.4, but with a different scale inside the window.



Figure 8.6: Same as Figure 8.4, but using two windows.

CHAPTER 9

STATUS OF THE EOS WORKSTATION AND TRANSITION TO PHASE-III

Upon completion of the Phase-II project, an elaborate software system exists whose components are in various stages of maturity. The target platform selected for the EOS workstation, the Stardent 3000, is a platform that in 1992 is obsolete. Given current innovation cycles in computer technology, this does not surprise. The situation is somewhat aggravated, however, by the fact that the manufacturing company, Stardent, has been liquidated and its assets have been turned over into a new corporation called Kubota Pacific Computers (KPC).

Many of the modules are implemented in a fairly mature fashion in spite of the fact that the goal of the SBIR Phase-II was to merely produce prototype software and have this converted into deliverable software under Phase-III.

Upon completion of Phase-II in December of 1991, an application was found for the EOS software in the Magellan project. Significant elements of the EOS workstation have become highly useful in processing Magellan radar images. This applies to the use of the softcopy light table, LT; the use of the digital elevation model software, CPS-3, and its related software to edit graphics elements, namely VX-Edit (see Figure 9.1); and various visualization tools to produce perspective views, etc.

Similarly the results of the EOS workstation are used internally at VEXCEL for processing various types of images under contract to the University of Colorado or under a NASA NRA project that is jointly being worked on with the University of California at Santa Barbara and a research group at the Jet Propulsion Laboratory.

The status of the software is such that the system as a whole is not deliverable because it lacks stability. It has the character of laboratory or prototype software.

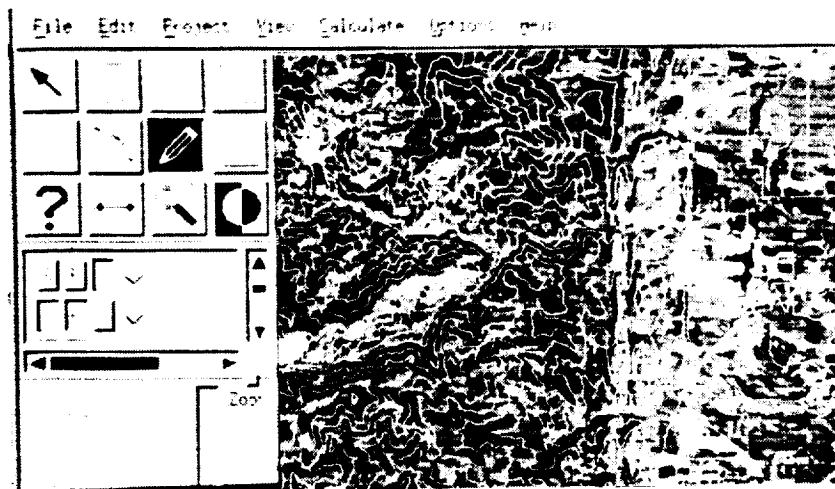


Figure 9.1: User interface for VX-Edit to simultaneously display an ortho-image and a contour map of topographic relief. Editing the contour map will automatically alter the underlying DEM from which the contour lines were generated.

Currently VEXCEL is negotiating with Kubota Pacific Computer Corporation a Phase-III effort, in which the entire suite of EOS Phase-II software and its enhancements that were funded outside of the SBIR funds are to be hosted on a generic graphics supercomputer of the next generation. This includes in particular the complete and mature light table LT software and the contour line editing software VX-Edit. This supercomputer is to be built around the 64-bit Alpha chip by Digital Equipment Corporation.

It is expected that late in 1992, or at the beginning of 1993, the EOS software will become available on this platform and will mature into a deliverable product. At that point the EOS software is poised to become broadly available for many years to come on a graphics supercomputer. Ultimately this same software could be available on

workstations from other vendors, since the software was developed as an "open system."

Note particularly, that the Magellan mission has expressed a need for some of the functions that have been implemented on the EOS workstation, particularly for LT, VX-Edit and the DEM software. Therefore, VEXCEL is currently porting some components of the EOS workstation from the Stardent 3000 to another target platform selected by the Magellan mission (Silicon Graphics IRIS or INDIGO workstation). This selection was made because it does support stereo viewing as a native capability. Delivery of that software to the Magellan-mission is planned for 1 October 1992.

CHAPTER 10

RECOMMENDATIONS AND OUTLOOK

10.1 Recommendations

With the advent of numerous commercially off-the-shelf image processing packages that are presented to the user by graphical user interfaces and run on generic RISC-based workstations, it becomes easier to configure a complex EOS image processing workstation because it can build on a rich range of functionalities that are already developed. This in turn is more easily linked than previously with commercially off-the-shelf geographical information system software. Most of GIS software currently supports also the simultaneous display of raster images as a native layer of a GIS. This again simplifies the implementation of basic GIS functions into a EOS work environment.

The core capabilities that are not broadly available and, therefore, need to be specifically added into an EOS environment are the abilities to deal with image cubes and to interact with large images, possibly multiple images, and stereoscopy. The EOS workstation contains these capabilities and they are the most important result of the Phase-II work.

It is recommended that the rapid innovation cycles in computer hardware be neutralized in such a manner that entirely standardized software is being made available with an X-Windows user interface, and adhering to standard UNIX and the C programming language. It is recommended that no specific image processing boards be applied but that one rely on the capabilities of generic graphics computer workstations. Computer workstations with 100 to 200 Mips and 150 MHz are available and make such specialized boards redundant.



In the dilemma between the PC and the RISC workstation, large data quantities enforce the use of a fast computer bus and, therefore, make the use of a PC at this time unlikely. However, since the EOS workstation concept should be just software, a concern for hardware should not exist. Hardware simply should be used as is available and needed.

Of course, the configuration of an EOS workstation that relies on commercially off-the-shelf software for image processing, for GIS, for relational data base and possibly for digital elevation modeling is still going to consume a certain investment. However, the set up of the EOS workstation should be such that the user has the option of using the capabilities of the workstation without third-party software. In addition, of course, personalized capabilities and application software should be integratable and it has been demonstrated that this is feasible and how it is to be done.

10.2 Outlook

Image processing in the EOS era will rely on many of the known and broadly available concepts of generic image processing, generic GIS and generic data base management. In addition there will be very specific demands on the ability to deal with image cubes, to perform stereoscopic analysis of multiple images and maintain a library of sensor models that permit one to relate world coordinate systems to image coordinates as new sensors evolve. We have demonstrated how such an approach works and have illustrated it with numerous currently available sensors.

The task that remains is to marry the innovative concepts developed for the EOS-workstation with the more mundane and traditional software packages commonly used in the pre-EOS era.

REFERENCES

Kober, W., J. Thomas, D. Meyer, F. Leberl (1988), *Requirements for an EOS-Oriented Workstation*. Proc. IGARSS 1988 Symp., Edinburgh, Scotland, 13-16 September 1988.

Maurice, K., W. Kober, F. Kruse (1991), *Visualization Environment for Imaging Spectrometry*. Presented at 8th Thematic Conf. on Geologic Remote Sensing, Denver, Colorado, April 29-May 1, 1991.

Millot, M., K. Maurice, M. Karspeck, W. Kober, F. Leberl (1991), *Generation, Visualization and Analysis of Image Cubes*. Presented at the Mission to Planet Earth Conference.

Thomas, J., R. Mullen, F. Leberl, W. Kober, J. Cimino (1988), *Concepts for Processing and Analyzing of Multiple SAR and Landsat Images*. SPIE Vol. 924, Recent Advances in Sensors, Radiometry, and Data Processing for Remote Sensing.



ACKNOWLEDGEMENT

The work under this SBIR Phase-II was closely supervised by numerous people at NASA's Jet Propulsion Laboratory. This initially was coordinated by Dr. Jobea Way and subsequently taken on by Dr. John Curlander. The help received from them was very crucial and greatly appreciated.

In addition a Scientific Advisory Board was assembled and consisted of a number of outstanding experts in remote sensing, namely Dr. Alexander Goetz of the University of Colorado at Boulder, Dr. Roger Hoffer of Colorado State University in Fort Collins and Dr. Roger Barry of the University of Colorado at Boulder. These individuals made themselves available in the early stages of the project to define their requirements for an EOS workstation and they were helpful with their advice in reviewing the requirements and specification documents.

The vendors of computer hardware to support the EOS workstation were initially VITEC, Texas and subsequently Stardent Computers, Massachusetts. The persons who enabled us to overcome many of the initial difficulties in dealing with this innovative hardware included Mr. Denis Colomb for whose help we are very grateful.

ATTACHMENT A

FUNCTIONAL REQUIREMENTS DOCUMENTS

- **Functional Description**
- **Environmental and Behavioral Models**

NASA - Jet Propulsion Laboratory
SBIR Phase II Grant

**EOS PROTOTYPE IMAGE ANALYSIS
WORKSTATION**

FUNCTIONAL DESCRIPTION

**Prepared by
VEXCEL Corporation**

2-July 1990

Boulder, Colorado

This represents a preliminary document for internal use by the Project

TABLE OF CONTENTS

1.	INTRODUCTION.....	1
2.	INPUT.....	1
3.	OUTPUT.....	4
4.	GEOMETRIC PROCESSING.....	5
5.	IMAGE CUBE GENERATION.....	19
6.	IMAGE CUBE ANALYSIS.....	23

— — — — —

1. INTRODUCTION

This document describes the EOS Prototype Image Analysis Workstation (EOS Workstation) functions from the user's point of view. It defines major functions available to the user, and options associated with each function. The format of the external interfaces is also described. We refer in this document to Data Flow and Context Diagrams from the Environmental and Behavioral Models, and also introduce additional figures.

2. INPUT

(See Figure 1 and Environmental Model: Context Diagram)

The EOS Workstation reads images and associated parameters from different sensors (see ENVIRONMENTAL MODEL: list of sensors) at different levels of geometric preprocessing. It loads vector representations of geographical features coming from an external Geographic Information Systems (GIS). We discuss third party software in a separate document (Software Management Plan) to specify the availability of a GIS.

2.1 Images

The user, during the image load phase, defines the number and names of the image bands to load and s/he defines the window to load by geographical/image coordinates of the four corners.

2.2 Auxiliary Data

The auxiliary data describe parameters specific to each sensor, the movement of the sensor, and information about the image acquisition such as the calibration data. Part of the auxiliary data may be recorded in the same physical media as the image. Therefore the loading of the auxiliary data typically happens during the image loading.

2.3 Digital Elevation Models (DEMs)

The EOS Workstation supports the DMA DTED USGS DEM formats. The user, during the DEM load phase, defines the window to load by cartographical/geographical coordinates.

1

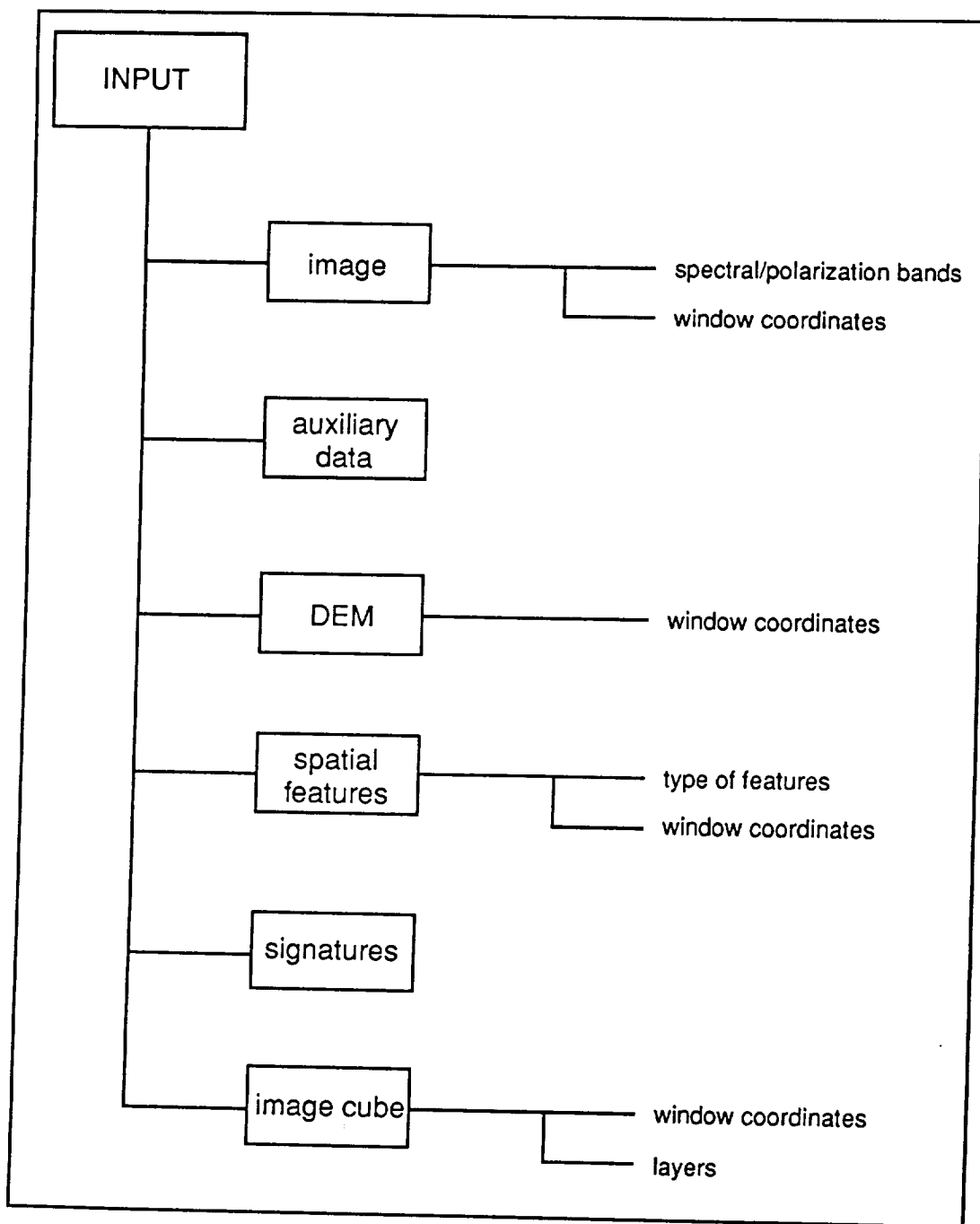


FIGURE 1
USER OPTIONS: INPUT

[illegible]

2.4 Spatial Features

2.4.1 Geographical Features

The user, during the load phase, selects the type of features and defines a spatial mask by cartographical/geographical coordinates of the four corners of an area of interest.

2.4.2. Image Features

An image feature contains the following information:

- Image name of image cube name and image cube layer name;
- Label and type;
- Feature image (cube layer) coordinates.

The user, during the load phase, selects the type of feature and defines a spatial mask by image coordinates of the four corners of the feature.

2.5 Signatures

The spectral/radar signatures include the following information:

- Object name;
- A variable number of pairs of data sets containing wavelength/frequency and reflectance/backscattering coefficients.

2.6 Image Cubes

The image cubes contain the following information:

- Cartographic parameters such as projection parameters, 4 corner coordinates of an area of interest.
- For each data layer: layer name, spectral/polarization band name, acquisition date.

The user, during the load phase, can define a rectangular window by cartographic/image coordinates and can also select the layers.

2.7 GIS

An external GIS is accessible through a data interface to extract a work area from, and load new or edited data back into, a GIS. See the "Software Management Plan" for further information on the GIS.

3. OUTPUT

(See Figure 2 and Environmental Model: Context Diagram)

The EOS Workstation can export image cubes and geographical features extracted from the image cubes and visualization files.

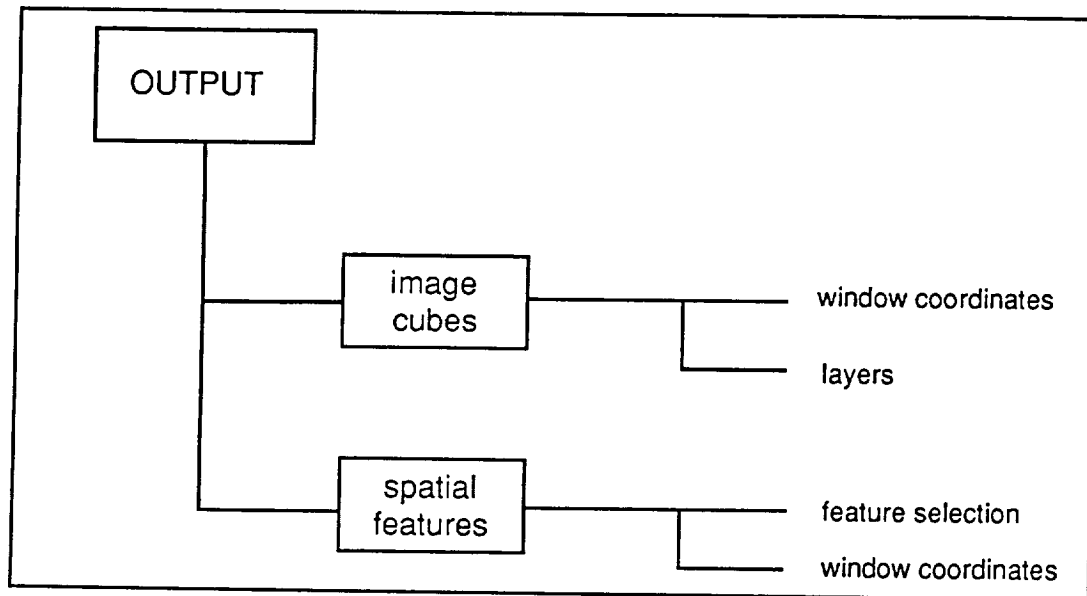


FIGURE 2
USER OPTIONS: OUTPUT

3.1 Image Cubes

The image cubes contain the following information:

- Cartographical parameters such as projection parameters, 4 corners coordinates of an area of interest;
- For each layer: layer, name, spectral/polarization band name, acquisition date.

The user, during the output/storing phase, can define a rectangular window of interest by cartographic/image coordinates and can also select the individual data layers for output.

3.2 Spatial Features

3.2.1 Geographical Feature

The user, during the output/storing phase, selects the type of features and defines a spatial mask by cartographic/geographic coordinates of the four corners

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

of an area of interest.

3.2.2 Image Feature

The image feature contains the following information:

- Image name or image cube name and image cube layer name;
- Label and type;
- Feature image (cube layer) coordinates.

The user, during the output/storing phase, selects the type of feature and defines a spatial mask by image coordinates of the four corners of an area of interest.

3.2.3 GIS

Data in the EOS Workstation can be transferred to an external GIS. See the "Software Management Plan" for more information.

4. GEOMETRIC PROCESSING

(See Behavioral Model DFD bubble 10)

Geometric processing is an important step for the generation of image cubes containing images from different sensors, from the same sensor but acquired with different geometric parameters or at different dates and at differing resolution. Several steps are involved in geometric processing:

- (a) Acquisition of common points between the image and the reference data set (either another image or data in the object/ground coordinate system);
- (b) Modeling of the mapping between the image and the reference (image/ground);
- (c) Resampling of the image;

But it also includes miscellaneous functions such as:

- (d) Geometric processing of features;
- (e) Perspective view generation;.

4.1 Coordinate Systems

We support the following coordinate systems:

- (a) Local display coordinates, in pixels on the display monitor;

- (b) Image coordinates in a sensor-specific image system such as cross-track/along track or pixel line and sample number;
- (c) 3-D cartesian object coordinate system with the origin at the center of an area of interest, XY in a user-specified plane system, and Z along the local zenith;
- (d) Geographic coordinate system with latitude, longitude and height above a reference ellipsoid;
- (e) Cartographic map coordinate system, the Universal Transverse Mercator System (UTM);

Additional coordinate systems may be added as needed.

4.2 Acquisition of Image Features

(See Behavioral Model: DFD bubble 10.5)

"Acquisition" is the interpretation and tracing of imaged features from an image in the computer, either automatically or manually. In the current context of geometric processing, "acquisition" refers to the collection of image data to support a geometric transformation of an image to the geometry of a map or another image.

Acquisition from a map, for example of Ground Control Point (GCP) in cartographic/geographic coordinates, is not addressed explicitly in the workstation. It is assumed that the user has available the cartographic/geographic coordinates of each GCP from outside sources. Addition of a digitizing tablet is feasible to support the digitization of map features but is currently not specifically planned.

4.2.1 Mono-Acquisition

(See Figure 3)

We differentiate between the image-image and image-map cases. Both employ the same "acquisition" functions. The user has at her/his disposal a number of feature acquisition tools. In principle, a global "area of interest" is defined and kept as a so-called "overview image". Inside the overview image is a local "area of interest" for current processing.

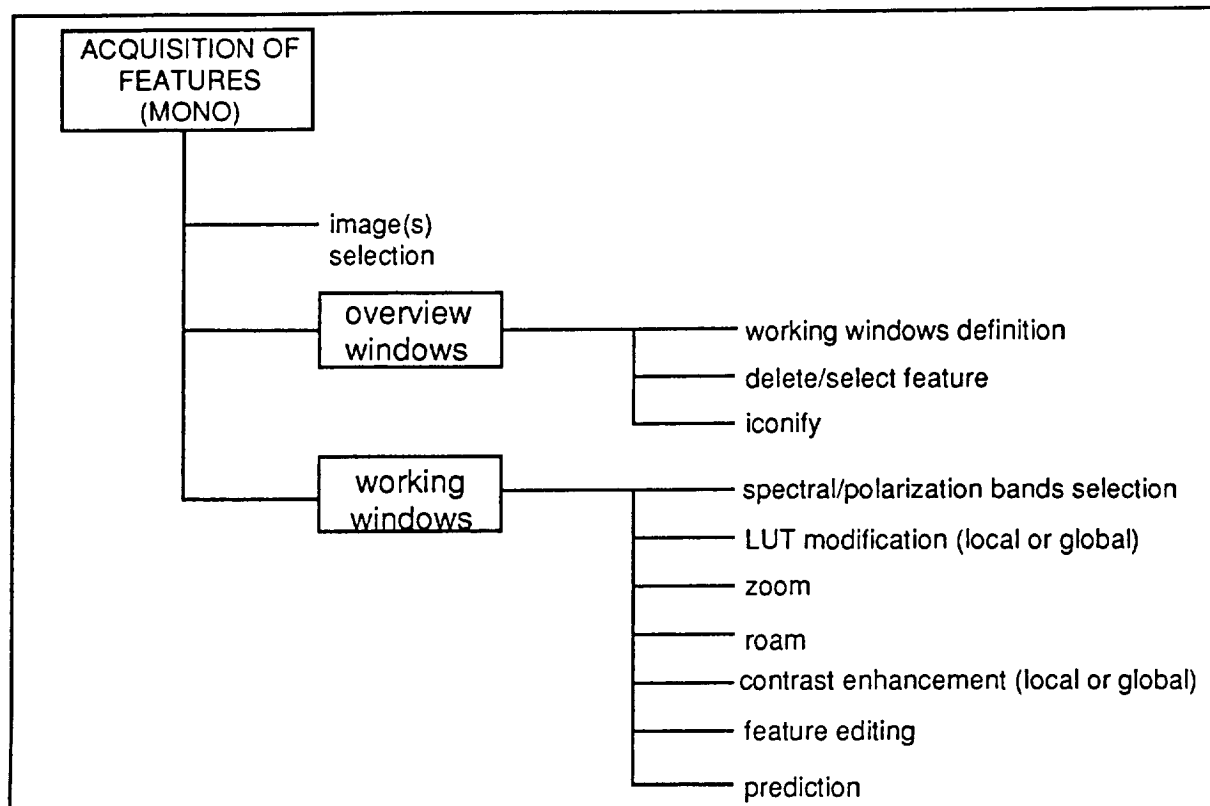


FIGURE 3
USER OPTIONS: ACQUISITION OF FEATURES
(MONO)

For geometric processing, we can deal with two images which are presented in an overview and with enlarged "working windows".

- (a) Two Overview Windows contain a subsampled version of two images showing the data points already acquired and the current location of the "working windows". These windows show the current status of acquisition. Inside the overview windows the user can define the location of the "working windows" and can also select or delete a feature. These windows can be "closed".
- (b) Two Working Windows containing the current "area of interest" at the user's specified resolution. The user can employ various operations as described in the following items.
- (c) Change the displayed spectral/polarization band(s). The selection is performed using band name(s) or dynamically displaying a band subset previously defined. The selected bands are displayed in the overview windows.

- (d) Modify globally or locally the intensity range for each window. Once the user has defined graphically the area to be processed, s/he can choose the default adjustment or modify dynamically the histogram of each image (3 bands for color visualization or 1 for B/W).
- (e) Zoom or unzoom. The outlines of the working windows are changed in the overview windows as one zooms/unzooms.
- (f) Roam in the displayed windows using window side bars. The outlines of the working windows are also changed in the overview windows.
- (g) Enhance the contrast.
- (h) Edit Features, for example:
 - Acquire a pair of features. During the acquisition, the cursor is active or passive, indicating that the current location is recorded or not. While the cursor is in active mode, the user is able to remove the point(s) previously defined. Once a feature is acquired in one image the corresponding displayed image is centered at the feature location and the user is asked to acquire the matching feature in the other image or to input the geographical/cartographical coordinates for the image-map case. S/he can always abandon the acquisition of a particular feature and label an acquired feature.
 - Select a feature. The user indicates in the subsampled windows or in the working windows the selected features by positioning the cursor on the feature or by defining a window containing the feature of interest. Once a feature is selected the user is able to modify (inside the working windows) the location and label or delete it.
- (i) Estimate the transformation (mapping) in a dynamic manner, using the acquired features and a low order polynomial mapping of the correspondence between the two images (called the PREDICTOR). The user uses the predictor to display the corresponding window of a displayed window. (Otherwise, correspondence would not be established.)

4.2.2 Stereo-Acquisition

(See Figure 4)

Matching of 2 images can also be based on stereoscopic viewing. The functions provided to the user are essentially the same as for mono-acquisition but with ad-

dition of same stereo specific functions.

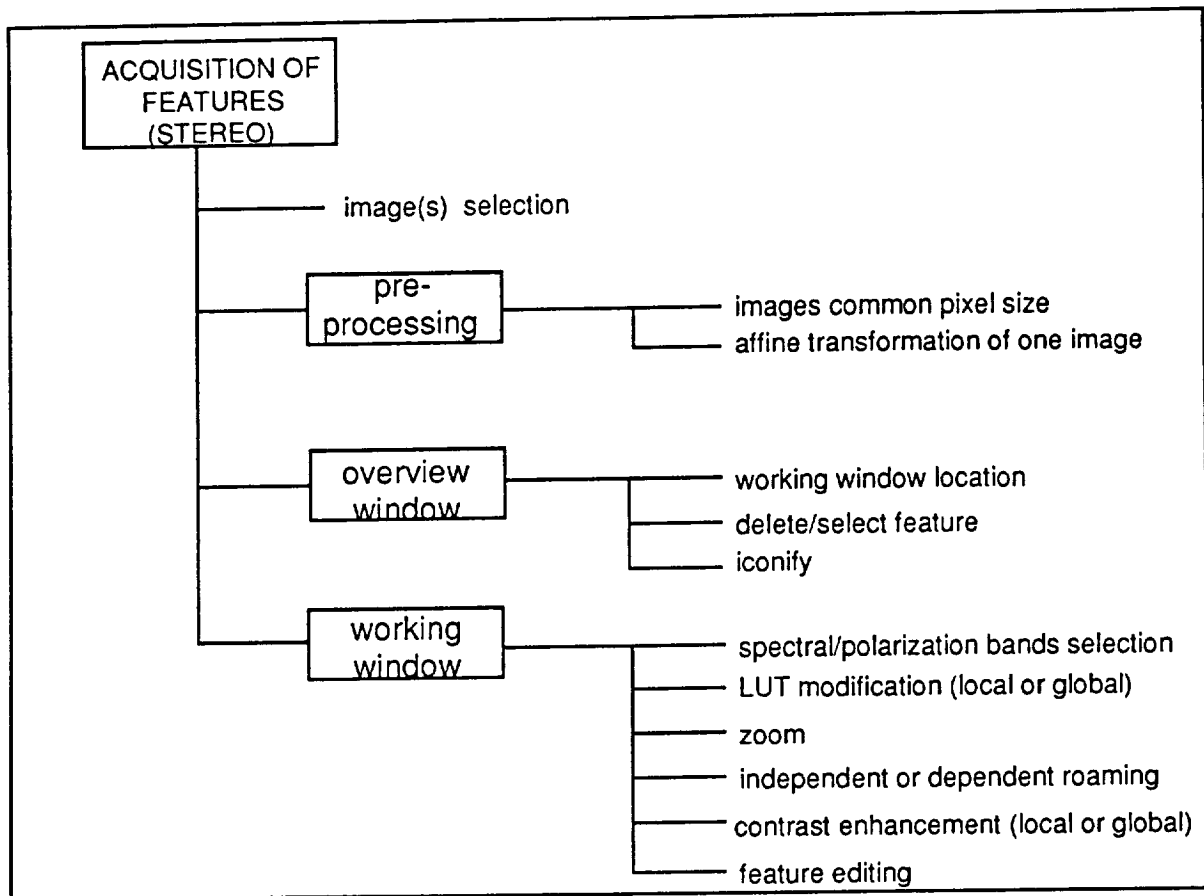


FIGURE 4
USER OPTIONS: ACQUISITION OF FEATURES (STEREO)

(a) Stereo Preprocessing

- The user defines a common pixel size between the 2 images to be matched, and the images are displayed with this new pixel size.
- The user defines an affine transformation of one image onto the other, using the MONO-ACQUISITION. One of the images is displayed with the transformation applied.

(b) Create Overview window.

One subsampled window containing a subsampled version of the two images, showing the points already acquired and the current location of the working window. This window represents the current status of acquisition. Inside this window the user can:

— — — — —

- Define the location of the working window.
- Select or delete a feature.
- Iconify it.

(c) Create Working Window.

A working window contains the current area of interest at the user specified resolution. The user can apply various functions within a working window.

(d) Change the displayed spectral/polarization band(s).

The selection is performed using band name or dynamically displaying a band subset previously defined. The selected bands are displayed in the overview window.

(e) Modify globally or locally the intensity range for each window. Once the user has defined graphically the area to be processed, s/he can choose the default adjustment or modify dynamically the histogram of each image (Two images for B/W and 6 images for color).

(f) Zoom or unzoom.

The outlines of the working window are also changed in the overview window.

(g) Roam in the displayed window. The images can be moved together or independently. The outlines of the working window are also changed in the overview window.

(h) Enhance the contrast.

(i) Acquire a pair of features.

During the acquisition, the cursor is active or passive. While in the passive mode, the parallax between the two representations of the cursor is fixed and they move by the same amount. Once the user decides to acquire a left or right image feature s/he can choose which cursor representation is fixed and which one is moving. S/he can always abandon the acquisition of this particular feature and label the acquired feature.

(j) Select a feature.

The user indicates in the overview or working window the selected feature by positioning the cursor on the feature or by defining a window containing

the feature of interest. Once a feature is selected the user can modify (inside the working window) the location and label or delete the feature.

4.3 Geometric Modeling

(See Behavioral Model: DFD bubble 10.2)

The level of geometric modeling applied to an image depends on the geometric level of the input image and the amount of information available on the sensor. We differentiate between two different models of the mapping between one image and another image, or between image and map/GIS:

(a) Physical Model

Data represent the geometric effect of the sensor and of the platform so that a model of physical phenomena is found.

(b) Empirical Model

A general estimation of the mapping is built without considerations regarding the sensor or the platform. This is sometimes referred as "warping" or "rubber sheeting".

4.3.1. Physical Modeling of Geometry

(See Figure 5)

Ground Control Points (GCP) or/and Image Control Points (ICP⁽¹⁾). If a user chooses not to use GCP or ICP, s/he performs a so-called System Correction (dead reckoning).

(a) Definition of parameters:

(a1) Geographical parameters of the "anchor points" or "grid". They define the location and the projection used. Anchor or grid points are auxiliary points for a geometric image transformation, at regular intervals in the output or input image. Actual resampling is done with a simple, bi-linear transformation inside a grid mesh. The grid or anchor points are computed with the best available model of the imaging process.

(a2) Grid mesh.

This defines the distance between grid nodes and is directly related to the modeling accuracy.

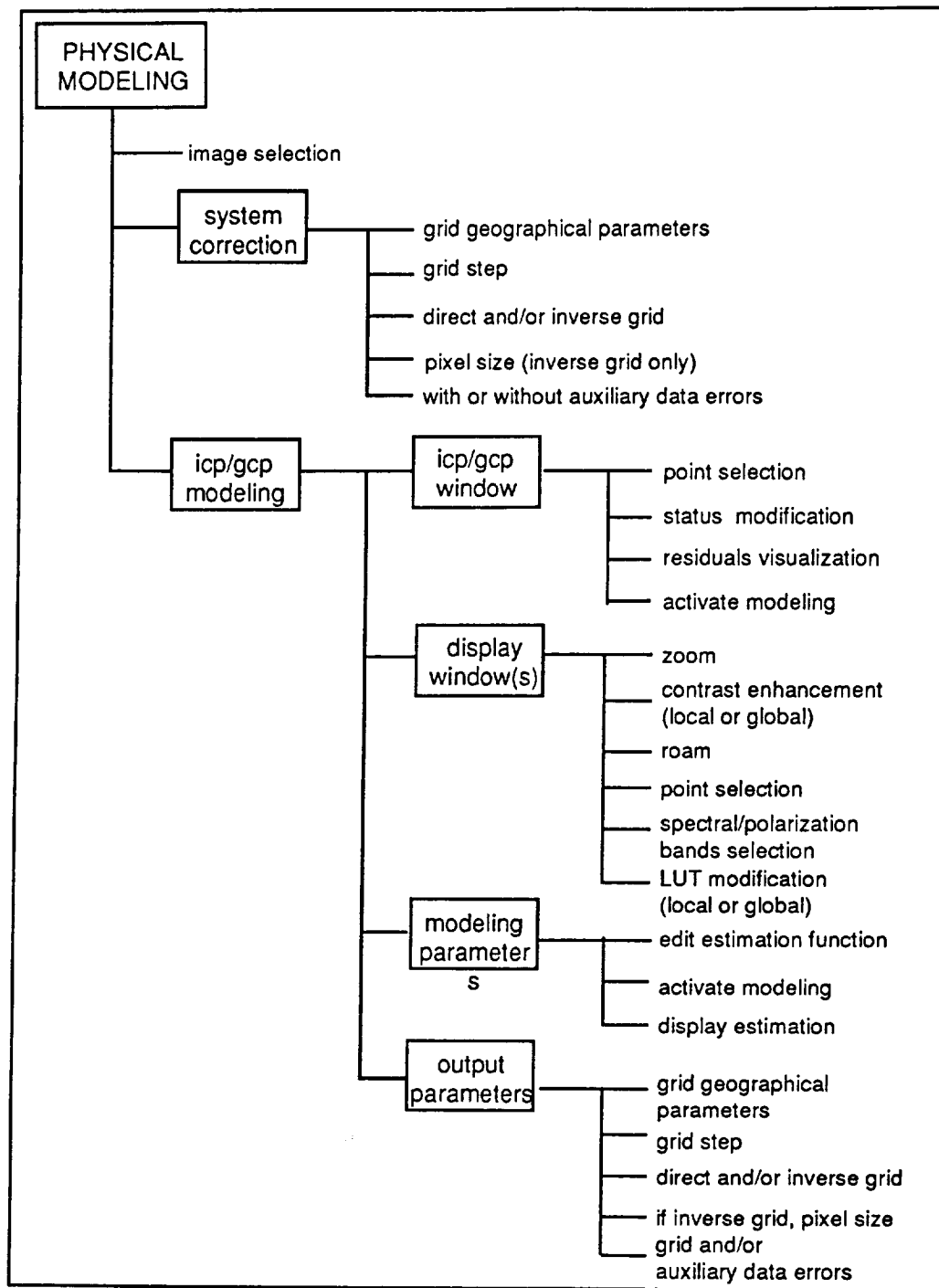


FIGURE 5
USER OPTIONS: PHYSICAL MODELING

- (a3) Direct and/or inverse grid. The direct grid models the mapping from the image to the ground (regular grid in the input image), the inverse grid models the mapping from the ground to the image (regular grid in output image).

- (a4) Pixel size. In the case of an inverse grid the user specifies the pixel size of the resampled output image.
- (a5) Auxiliary data corrections. The user may use previously estimated corrections to the auxiliary data to generate the grid and therefore obtain a more accurate mapping between the image and the ground.

(b) Tools

If the user chooses to adjust the flight parameters using GCP or ICP, s/he has at her/his disposal a number of tools as discussed in the following.

- (b1) A window exists which contains the list of ICP/GCP already acquired. The points are represented image(s) and/or ground coordinates, geometric residual and a label. A "residual" is the difference between the actual ICP/GCP position in the image and that which results from the model of the imaging process.

The user can now:

- Select a particular ICP/GCP point.
- Update the modeling status of this point:

- (i) Active point: the point is used for the modeling computation.
- (ii) Check point: the point is not used for the modeling but residuals are computed.
- (iii) Passive: the point is ignored.

- Residuals visualization. Computation and display of the overall and individual residuals on both active and check points. When both ICPs and GCPs are used simultaneously, they have a different representation.
- Activation of modeling. Once modeling is activated the user cannot modify the modeling parameters; after completion of modeling, the point list, residuals and modeling parameter files are updated.

- (b2) Display window(s) can be used by the user namely:

One window (if image-map or image ground modeling) or two windows (if image-image modeling or image-image-ground modeling) containing the image(s). These windows, display the location of ICP/GCP and optionally the residual vector associated with active and check points. The user can perform some image processing functions like:

- Modify the resolution of each image,

- Globally or locally modify the intensity range,
- Enhance the contrast,
- Change the displayed window.

(b3) One window is available to describe the modeling function. The default set of parameters to be estimated, and the analytical form of the function describing each parameter are displayed for each sensor. The user can:

- Add or remove parameters from the estimation.
- Choose an alternative analytical description.
- Activate the modeling.

(b4) A window exists to define the modeling outputs. The user can select to use either the current estimate of geometric parameters, one or two grid mesh representations of these parameters, or both. If a grid mesh representation is chosen the user defines:

- The geographical parameters of the grid points to define their location and the projection used.
- Grid mesh size to define the distance between grid nodes.
- Whether a direct is used or inverse grid. The direct grid models the mapping from the image to the ground, the inverse grid (resampling) models the mapping from the ground to the image.
- Pixel size. In the case of an inverse grid (resampling) the user specifies the pixel size of the resampled image. In "forward sampling", i.e. direct grid, the input image defines pixel size.

4.3.2. Empirical Modeling

(See Figure 6)

The user interface is similar to the PHYSICAL MODELING, but the user chooses the functional description of the mapping and not the functional description of the platform parameters. In the case of image to image registration, the user has the option to activate a command to monitor the registration action in stereo.

4.4 Geometric Processing of Features

(See Figure 7, and DFD bubble 10.1)

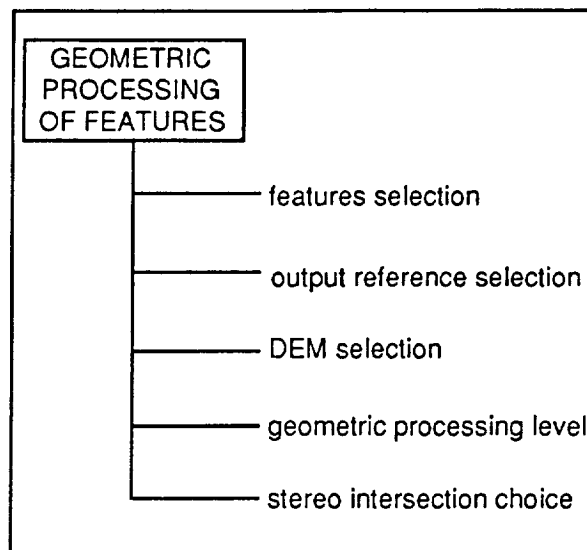


FIGURE 7
USER OPTIONS: GEOMETRIC PROCESSING OF FEATURES

This function is used in two different cases: to project external (map of GIS) features into the image coordinate system, or to compute the geographic coordinates of features acquired in the original input images. For this purpose, the user has to make a number of selections.

- (a) Choose the features to be geometrically processed by their label, their geographical area, their date of acquisition or/and image names.
- (b) Choose the output reference coordinate system as ground or image. The ground system is specified as either geographic or cartographic, the image system by image name.
- (c) Select a DEM if required and available. The registration between input image and DEM is verified (see item....).
- (d) Select the geometric processing level. The system checks automatically if a geometric transformation is known between the feature (i.e. image input) and the output reference and will indicate to the user the geometric processing level available. If several levels are possible, the user is asked to choose one of them and to confirm the processing. "Level" is the complexity of the mapping as represented by the number of unknown parameters one can determine.

- (e) Choose the stereo intersection option. In the case of features acquired in a stereo pair, the user is specifically asked to indicate that s/he wants a stereo intersection to be performed. If so the default output coordinate system is the ground. The system checks that this feature was acquired in both images of the stereo pair and also if a coordinate transformation exists between this stereo pair and the ground.

4.5 Perspective View Generation

(See Figure 8, and DFD bubble 10.4)

For this function the image and the elevation model must be co-registered (see item....). The perspective view generation is done in several steps as follows:

:

- (a) Image selection. An image may be selected by image name or from a given list of images, selecting one or more of the following user defined criteria: sensor, acquisition date, geographical coordinates.
- (b) DEM selection. The DEM may be selected by DEM name, or from a given list, by requiring that user defined geographical coordinates are inside a DEM.
- (c) Feature selection. Features may be selected by feature name or from a given list of names using certain criteria such as type, label, geographical coordinates.
- (d) Selection of perspective view parameters. The elevation model is displayed at a low resolution and the user can then define the viewing parameters in two ways:
 - Real time manipulation of the elevation model using the knob box and moving the viewing position and imaging parameters.
 - Keyboard input of the viewing/imaging parameters and subsequent check of the elevation model using these parameters.
- (e) Generation of a low resolution perspective view image. This low resolution image is used to check and verify that the final image corresponds to the user requirement.
- (f) Generation of a final perspective view for storage in a file.



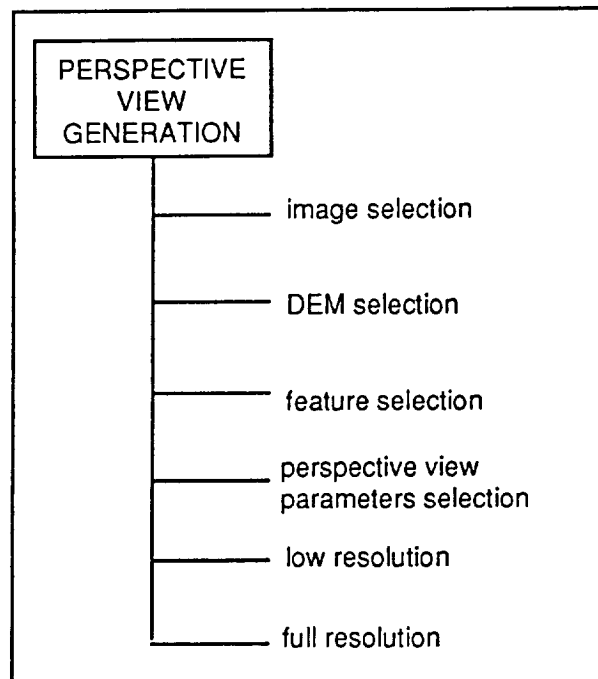


FIGURE 8
USER OPTIONS: PERSPECTIVE VIEW GENERATION

4.6 Resampling

(See User Options: Figure 9, and DFD bubble 10.3)

The user is asked to provide a number of parameters for final resampling of an input image.

- (a) Name of the input image. Optionally the user is able to pop up a window containing the existing list of images and select directly the desired image name or select by sensor name, acquisition date and geographical location.
- (b) Resampling method. The user is asked to choose between nearest neighbor, bilinear or bicubic resampling name. The user is shown the existing transformation to other coordinate systems; the user then chooses the required output system.
- (c) At the user's request, a list of DEM's intersecting the imaged area is displayed with the 4 corners coordinates of each DEM. The user chooses the DEM to be used.
- (d) In the case of SAR images with a DEM, the user has the choice between resampling via anchor points and resampling of each pixel individually

1. *What is the main purpose of this study?*

2. *What are the research objectives?*

3. *What is the significance of the study?*

4. *What is the scope of the study?*

5. *What are the limitations of the study?*

6. *What is the methodology used in the study?*

7. *What are the results of the study?*

8. *What are the conclusions of the study?*

9. *What are the implications of the study?*

10. *What are the future research directions?*

which permits one to remove effects of possible layovers and refined local transformation due to foreshortening.

- (e) The window to be generated. The user chooses a window in the output coordinate system reference or in the input image when s/he requires only part of the original image to be resampled.
- (f) A scale factor. The default output image pixel size is already defined in the grid definition (item number ...), but the user can provide a scale factor to apply to the pixel size so that output pixel spacing differs from the default value in the anchor pixel file.

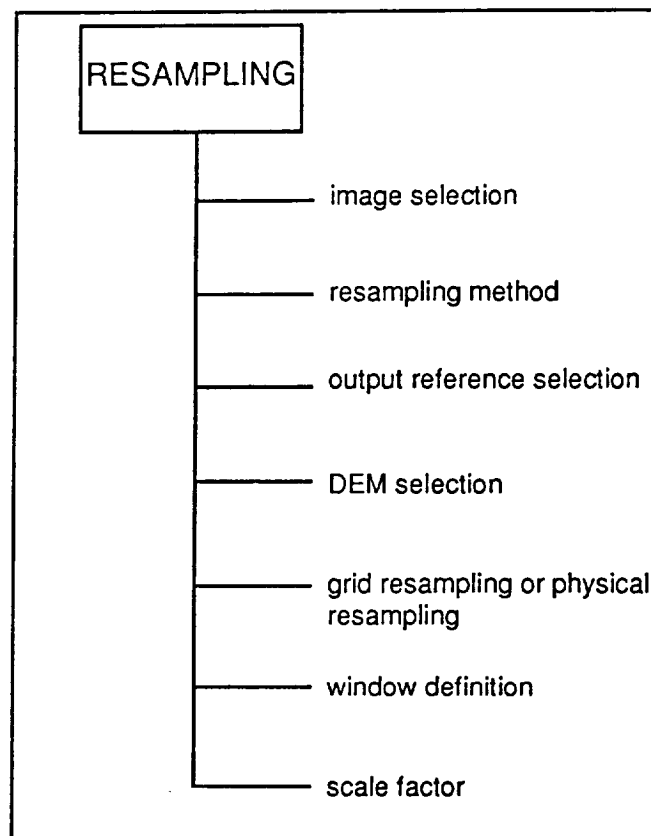


FIGURE 9
USER OPTIONS: RESAMPLING

5.0 IMAGE CUBE GENERATION

(See Figure 10, and DFD bubble 5)

5.1 Data Types in a Cube

The main attribute of the image cube is its spatial and spectral definition. This consists of different data such as:

- (a) Geocoded or image-coded (i.e. image-to-image registered) images from different sensors acquired at different dates.
- (b) Spectral multiplicity;
- (c) Spatial extent of pixel lines and samples per line;
- (d) Digital Elevation Model (DEM);
- (e) Theme data in image form (so-called "Features")

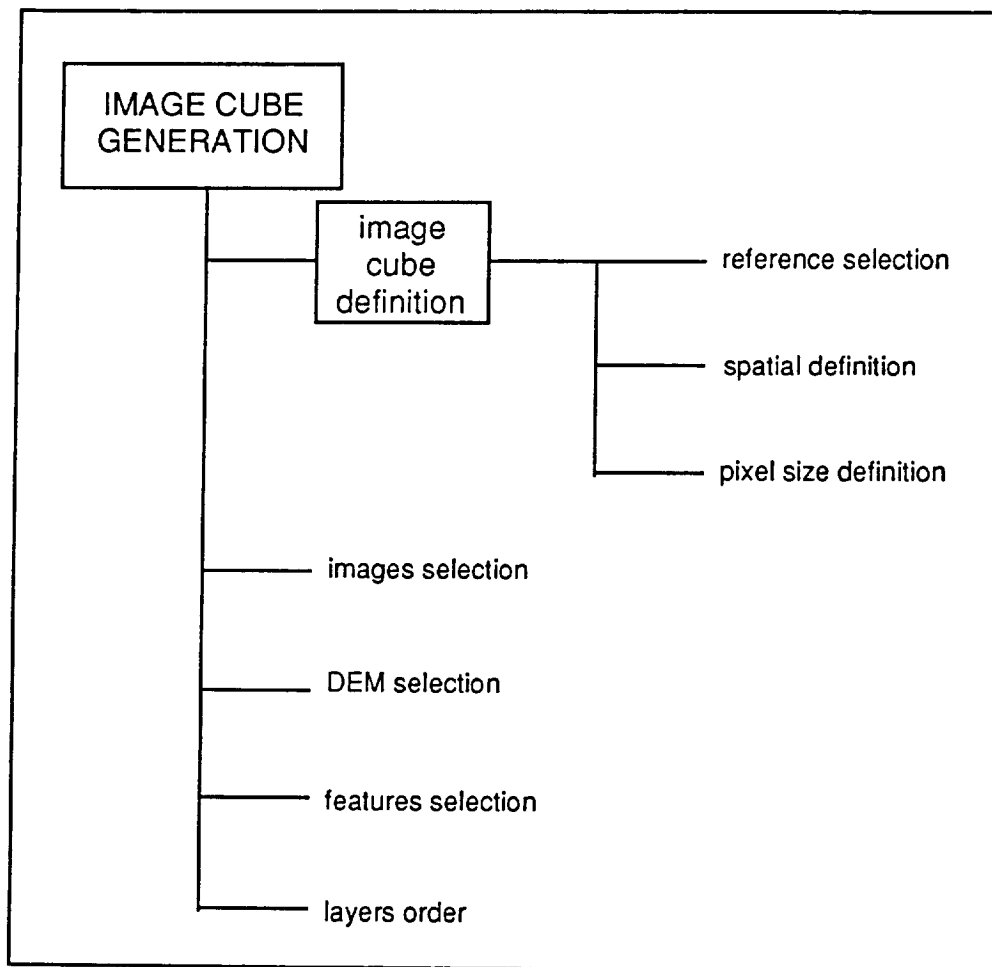


FIGURE 10
USER OPTIONS: IMAGE CUBE GENERATION

5.2 Building of Cube

The user can perform operations on the cube before an analysis begins. This includes the following:



(a) Define the image cube.

- Definition of coordinate system either on the ground or in an image. The system is the link among all the layers and each layer must be in this system.
- Area definition in the reference coordinate system. For a ground system, the user is asked for the geographic or cartographic coordinates of the 4 corners of an area of interest. For an image system, the user is asked for the image coordinates of the 4 image corners. Alternatively, the user is able to define the area of interest in the reference image or a geocoded image by drawing a rectangle inside a display window containing the reference image. When the reference is the ground, the user must provide the name of a geocoded image containing the potential area of interest, or define a geographical/cartographical rectangle containing the area and choose one image from the list of images intersecting this area definition.
- Pixel size. A common pixel spacing is used for all the images in the image cube.

(b) Image Selection

Once the area is defined the user requests a list of images satisfying the area definition and providing the coordinates transformation. The user chooses the images to be included in the image cube.

(c) DEM selection

At the user's request, the system checks the existence of a DEM covering the area of interest. If such a DEM is found, it is formatted as an image and resampled to the previously defined pixel size.

(d) Feature selection

At the user's request, the system checks the existence of geographic or image features inside the area of interest. The user is able to select features by label, and to create sets of features to be formatted as a unique image.

(e) Sequence of Component Image Layers.

Once the layers are selected and generated, the user defines the sequence of layers in the cube. The user can attribute a number to each

layer in the layer list, alternatively the user can display in one window the existing and in another window the ordered image cube. The user displays sequentially the layers in the existing image cube window and stops the animation at the layer to be transferred to the ordered image cube window.

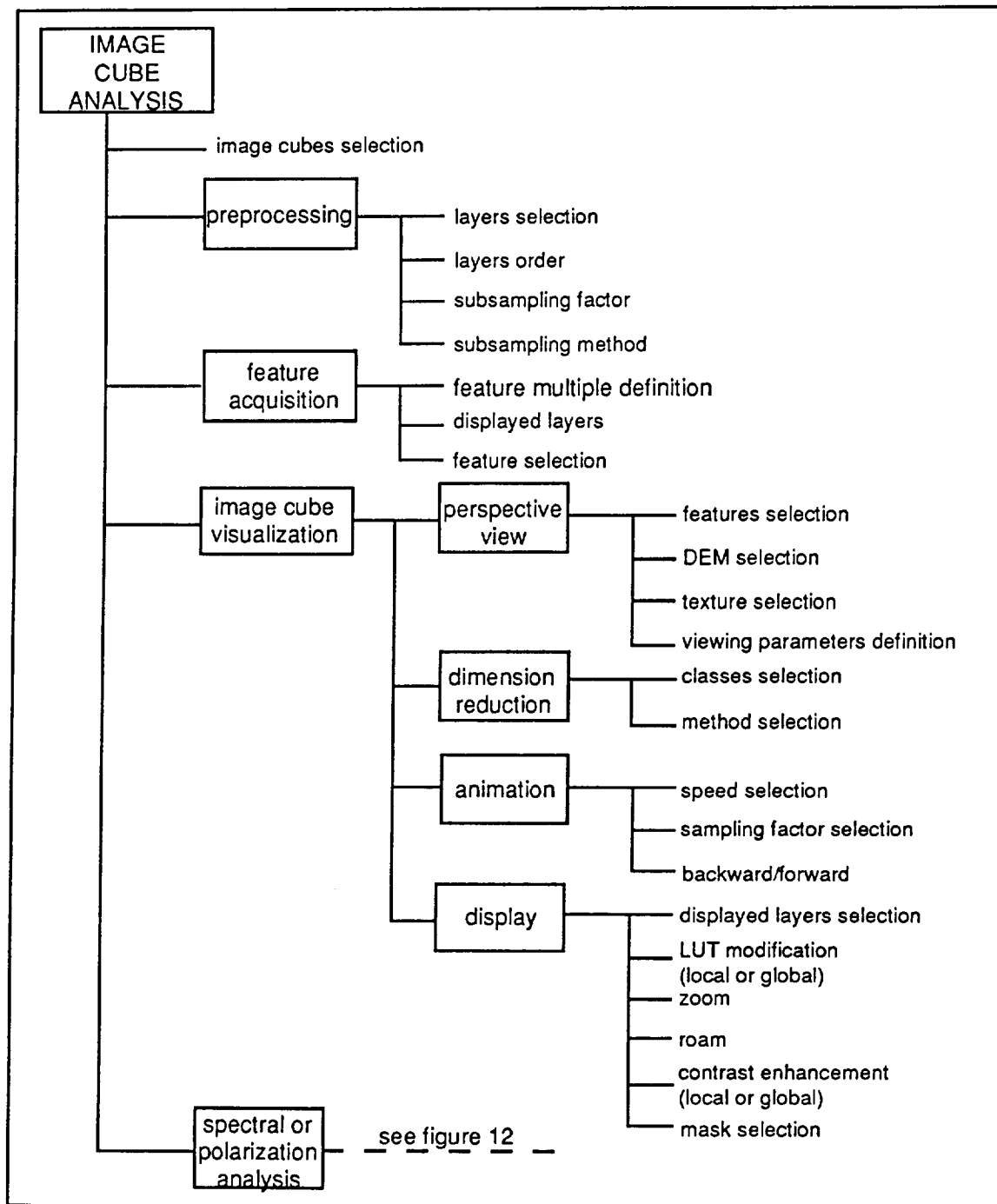


FIGURE 11
USER OPTIONS: IMAGE CUBE ANALYSIS



6.0 IMAGE CUBE ANALYSIS

(See Figure 11, and DFD bubble 9)

The image cube analysis is the most open part of the system. The functions described below represent a minimum set, but do not cover all potential functions useful for image cube analysis.

6.1 Preprocessing

A number of preprocessing steps has to include the following:

- (a) Selection of the layers required for the analysis. The user chooses the layers from a list containing the layer names or by use of a window containing the existing image cube and another window containing the new image cube. The user moves a layer from one window to the other and may display sequentially the images in each of the windows. Optionally the selection is done by sensor, spectral band and date of acquisition.
- (b) Redefinition of the order of image layer.
- (c) Modification of the pixel size. A smaller pixel size may reduce the processing time; therefore a capability exists to downsample an existing cube.
- (d) Selection of an interpolation method between image layers.

6.2 Feature Acquisition

(See DFD bubble 9.2)

Feature acquisition in an image cube has all the functions also offered in feature acquisition of the geometric processing sequence. Additional capabilities are:

- (a) Definition of the same feature in different layers of the image cube. Therefore the feature has different cartographic/image coordinates depending on the image cube layer considered.
- (b) Opening of several windows, containing different layers, showing dynamically the definition of the feature in several layers.
- (c) Sequential display of the image layers with the features overlaid.
- (d) Display of the location of the already acquired features in an overview



window.

6.2 Image Cube Visualization

(See DFD bubble 9.4)

(a) Perspective View Generation

The constraint exists that the DEM must be a part of the image cube (the elevation model can be something else than elevation, for example moisture or another image).

(b) Stereo Perspective View

The user can define perspective views for stereo visualization.

(c) Include Features in Perspective View

As an option, the user includes features in the perspective view. The user has a list of features included in the area of interest and can select a feature by name, type, label, or cartographic coordinates.

(d) Data Compression

The user reduces the number of spectral/polarization bands from n to m . The user can do this globally or can define areas to be used in the reduction algorithm.

(e) Display

A sequential display of an entire cube or display of two subsets of the image cube in two different windows is feasible. The user has control of the resolution, display speed, sampling and direction (forward or backward).

(f) Modify the display:

- The user can change the displayed layer(s). The selection is performed by using the layer name or dynamically displaying a band subset previously defined. The color assignment is done by specifying the layer names or visually by sequentially displaying the layers and selecting a layer for each color. Once a layer is chosen, it is displayed in another window with the corresponding color.
- Modify globally or locally the intensity range for each window. Once the user has defined graphically the area to be processed, s/he

chooses the default adjustment or modifies dynamically the histogram of each displayed layer(s).

- Zoom or unzoom.
- Roam the images in the displayed windows using window side bars.
- Enhance locally or globally the contrast of the displayed images.
- Use a visualization mask to selectively display part of the image cube. The mask is defined interactively by the user, or is an existing feature that the user chooses from the existing feature list. The choice is done by feature name, type, label or by inclusion in a rectangular window defined interactively.

(g) Visualize several image cubes. More than one cube can be displayed on one monitor.

6.4 Spectral or Polarization Analysis

(See Figure 12, and DFD bubble 9.3)

Some spectral or analyses are restricted to one sensor. An example is radar backscatter. For such analysis the system offers certain tools.

- (a) Select one or several signatures. The user can define a signature by material name or select the signatures from a list. The selection may also be done by materials defined in the wavelength/frequency range of the image cube sensor.
- (b) Display signatures. The user can sequentially display the selected signatures or display the selected signatures at the same time with different colors. The user chooses one signature by stopping the sequential display or by indicating it in the window containing the selected window.
- (c) Select spectral/polarization band(s). The user can choose the displayed spectral/polarization bands, using the signature. In the signature window s/he moves the cursor at a particular location and the nearest band is dynamically displayed, or s/he defines graphically one or several wavelength/polarization intervals, therefore creating a new image cube.
- (d) Display, acquire and select signatures.
 - Display the signature of a particular pixel. The user can move the cursor in the image window and dynamically display the corresponding signature. The user can record the location of a particular pixel and



build a pixel signature list for further investigation. The display of this pixel signature list is similar to the display of signature list.

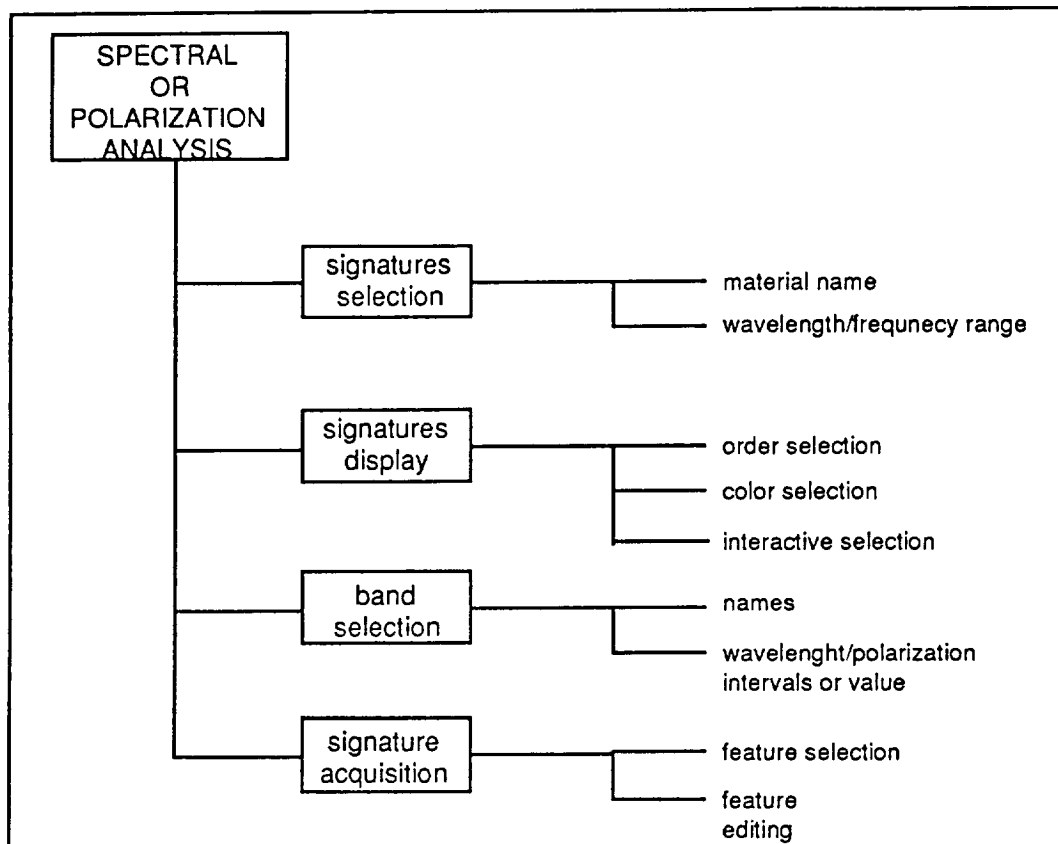


FIGURE 12
USER OPTIONS: IMAGE CUBE ANALYSIS
(SPECTRAL/POLARIZATION ANALYSIS)

(e) Display the signature of a particular line. S/he can acquire the line in the image window and dynamically store the corresponding signatures. The user can record the line location and build a line signature list for further investigation. The mean and envelop for each line is displayed, and the user then has the same tools as those provided with the display of signature list.

- Display the signature of a particular vector in the image cube. The user can define the polygonal outline of the area in the image window and, once the area is closed, store the corresponding signatures. The user can store the area and build an area signature list for further investigation. The mean and standard direction for each area are displayed, and the user then has the same tools as with the display of a signature list.
- Label pixels, lines or areas.

- Display the signature of an existing feature; the feature can be selected by feature name or from an existing feature list.

(f) Polarimeter

NASA's SAR polarimeter software may be incorporated into this range of capabilities.

(g) Slope-effect reduced images

Once a SAR image and DEM are registered, one has the capability to eliminate the terrain slope effect from the radar image gray values, assuming a backscatter function.

NASA - Jet Propulsion Laboratory
SBIR Phase II Grant

**EOS PROTOTYPE IMAGE ANALYSIS
WORKSTATION**

ENVIRONMENTAL AND BEHAVIORAL MODELS

**Prepared by
VEXCEL Corporation**

2 July 1990

Boulder, Colorado

This represents a preliminary document for internal use by the Project

TABLE OF CONTENTS

1.	INTRODUCTION TO THIS DOCUMENT.....	1
2.	INTRODUCTION TO THE ENVIRONMENTAL MODEL.....	1
3.	INTRODUCTION TO THE BEHAVIORAL MODEL.....	1
4.	COMMENTS.....	2
5.	THE ENVIRONMENTAL MODEL.....	2
6.	THE BEHAVIORAL MODEL.....	5

1. Introduction to This Document

The purpose of this document is to give a general description of the "Earth Observing System" (EOS) prototype image analysis workstation software ("EOS Workstation"). The document presents the current

- understanding of the EOS analysis software ("EOS Workstation");
- design of the EOS Workstation.

This document contains two models of the EOS Workstation:

- (a) Environmental model. This models the context of the system and describes what information enters from the external sources and what information is produced by the system.
- (b) Behavioral model. This presents the functional requirements and modifies those requirements of the EOS Workstation which we described in VEXCEL's quarterly Report #2 dated 16 November 1989.

Note that the current EOS Workstation analysis has matured from earlier efforts due to continued discussion and comparison with existing software and a better understanding of the user's needs.

2. Environmental Model

The "Environmental Model" contains:

- (a) The Statement of Purpose describing the overall purpose of the system.
- (b) The Context diagram, highlighting important characteristics of the system, such as:
 - Data received from and data sent to the outside world;
 - Data files shared between the system and the outside world.
- (c) List of Sensors. The list of sensors specifies the types of images the system will be able to read without specific user development.

3. BEHAVIORAL MODEL

The "Behavioral Model" contains:

- (a) High Level Data Flow Diagrams (DFD). They reflect the introduction of image cubes and highlight the potential use of a GIS.
- (b) Entity-Relationship Diagram. This gives a high level description of the relationship between the data stored in the system.

1

- (c) Data Dictionary. It describes the meaning of the flows and data files shown in the DFD and, when relevant, the units and range of the data.

The Functional Description is put into a separate document which puts the emphasis on the user interface and describes, when possible, the options the user will have.

4. COMMENTS

Clearly, the initial EOS Workstation will have capabilities that address only specific, limited needs. Revisions will exist to extend the system later to add new capabilities.

While this document defines the "EOS Workstation," we emphasize that this definition is subject to change.

EOS remote sensing will address almost all earth science fields, atmospheric and other sciences. The current EOS Workstation Project emphasizes some functionality that is expected to be common to geosciences dealing with the land, i.e., geology, forestry, land use, planning, etc. These fields have need for high resolution remote sensing data in image form. This may be a contrast to atmospheric or water-related sciences which may have important uses for non-image data.

5. ENVIRONMENTAL MODEL

5.1 Statement of Purpose

The purpose of the EOS Workstation is to provide EOS investigators with a versatile tool for rectifying, co-registering, and geocoding images from sensors in the pre-EOS era, for visualizing these images and their interrelationships, and for extracting features and interpretive data from the images for subsequent import to a Geographic Information System (GIS). It will also allow them to manipulate non-image information (spatial features, spectral or polarization signatures). Various types of imagery are integrated via *image cubes* consisting of co-registered images from different sensors or spectral bands. The image cubes and non-image objects extracted from the image cubes are available to other systems such as user-specific software and GIS.

5.2 EOS Workstation Context Diagram

Figure 5.1 presents the environment in which the EOS Workstation will operate. Note that a multiplicity of data types will be input into, and result from, the EOS Workstation.

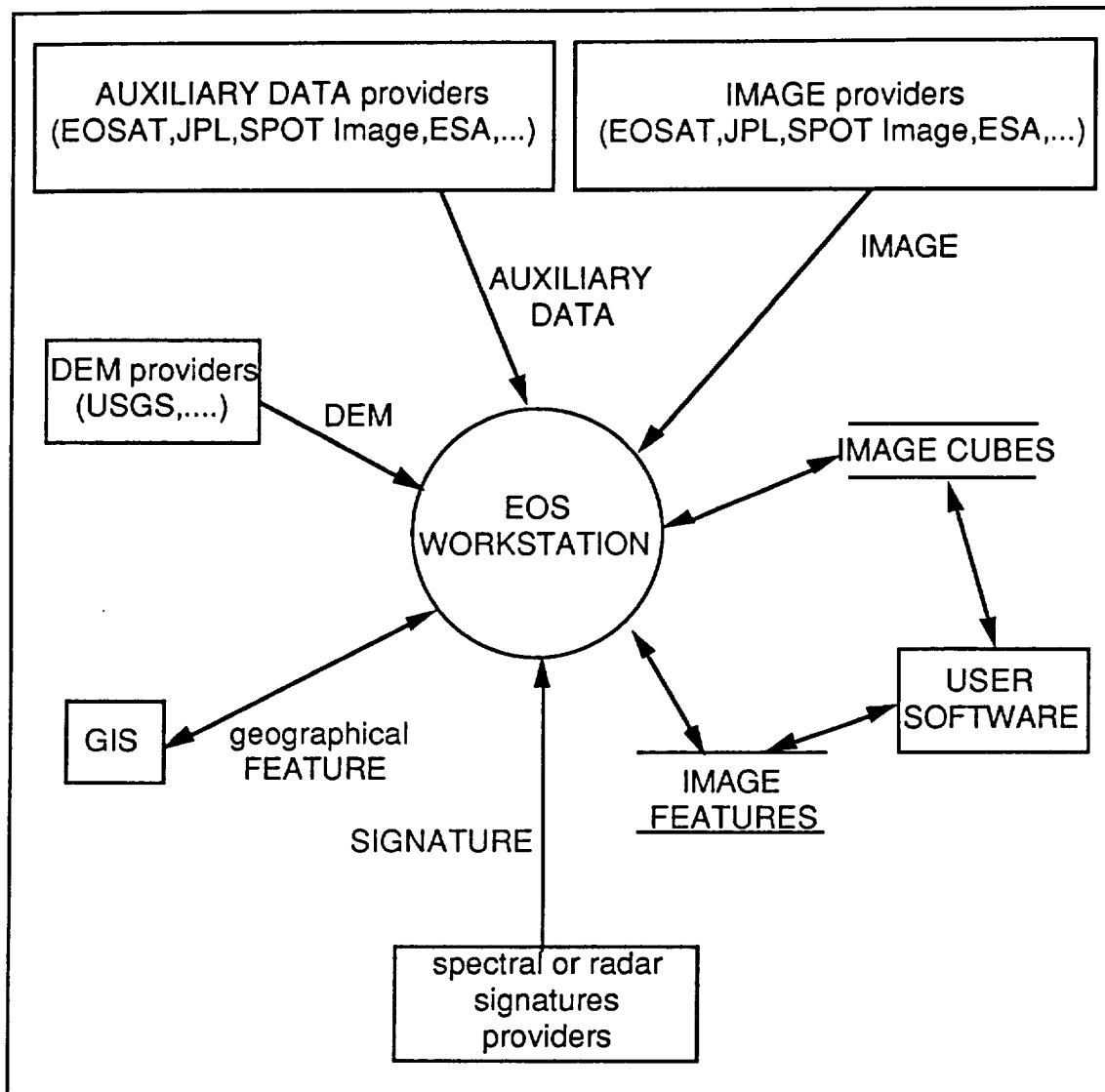


Figure 5.1: EOS WORKSTATION CONTEXT DIAGRAM

5.3 List of Sensors

The EOS Workstation will initially be capable of processing the following remote sensing image data:

- LANDSAT Thematic Mapper,
- SPOT Panchromatic Imager,
- SPOT Color Imager,
- AVIRIS Aircraft Sensor Imager,

- Multipolarization, multifrequency SAR from Aircraft,
- AVHRR from satellite.

Optionally, additional imagery may include:

- LANDSAT MSS,
- Aerial Photography,
- European ERS, and others.

The primary sensors are selected for the following reasons:

- LANDSAT TM provides 7 (6) spectral bands with multitemporal coverages and fairly systematic availability;
- SPOT Panchromatic Imager provides high resolution with a stereo capability and systematic coverage;
- AVIRIS is a simulation of the EOS-HIRIS capability;
- Multipolarization, multispectral aircraft SAR typifies future EOS-SAR data;
- AVHRR provides a global coverage of large areas and produces an understanding of multiple resolution data sets when seen in combination with other remote sensing data.

5.4 Other Data

From the Context Diagram, it is apparent that the EOS Workstation will have to be capable of working with:

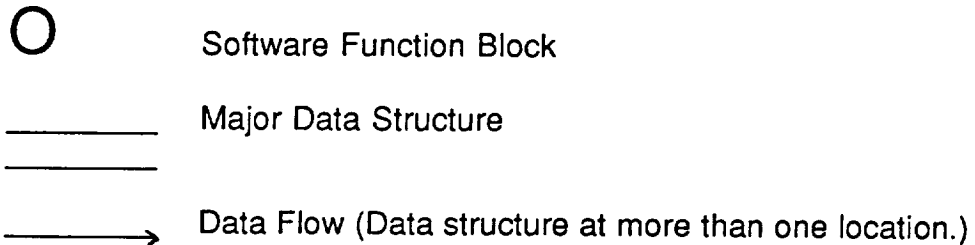
- Digital map data from a Geographic Information System;
- Digital elevation models and
- Various non-image data that accompany image data (so-called "auxiliary data") and
- Ground truth data such as spectral and other object signatures (backscatter, etc.)



6. BEHAVIORAL MODEL

6.1 Data Flow Diagrams

We describe in a few data flow diagrams (DFDs) how the Workstation software will be organized into major function blocks. The following conventions are used:

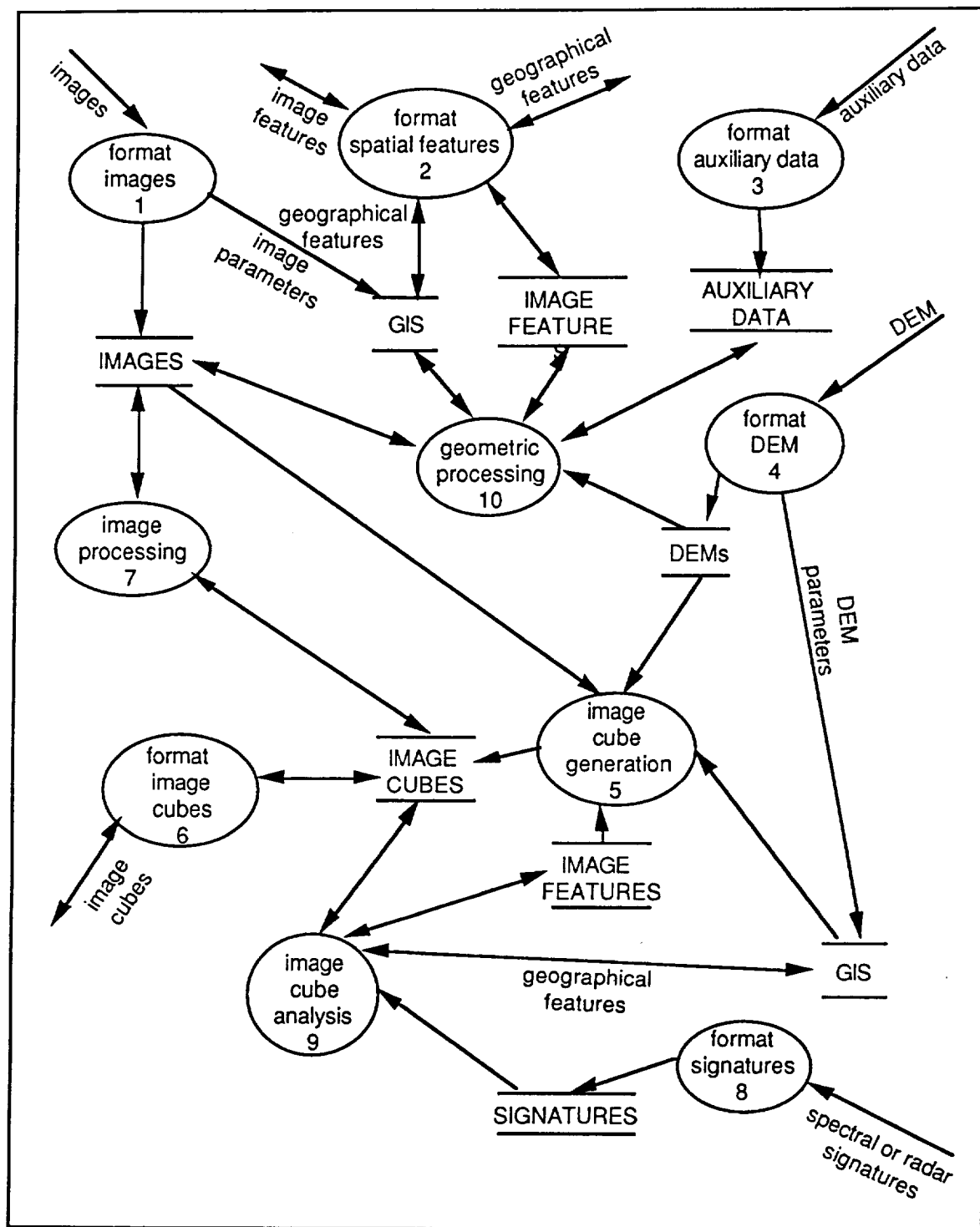


DFDs speak for themselves. Individual concepts are explained in the "Data Dictionary."

Note that in the initial implementation, a Digital Elevation Model (DEM) is only going to be used if it is externally provided. Resources may not exist to implement a stereo measurement capability to actually create a DEM from SPOT, radar or photographic images.

Stereo viewing will be implemented, as will be discussed later in the "Functional Specification." However, this will mainly serve the purpose of supporting the interactive analysis by the image interpreter.







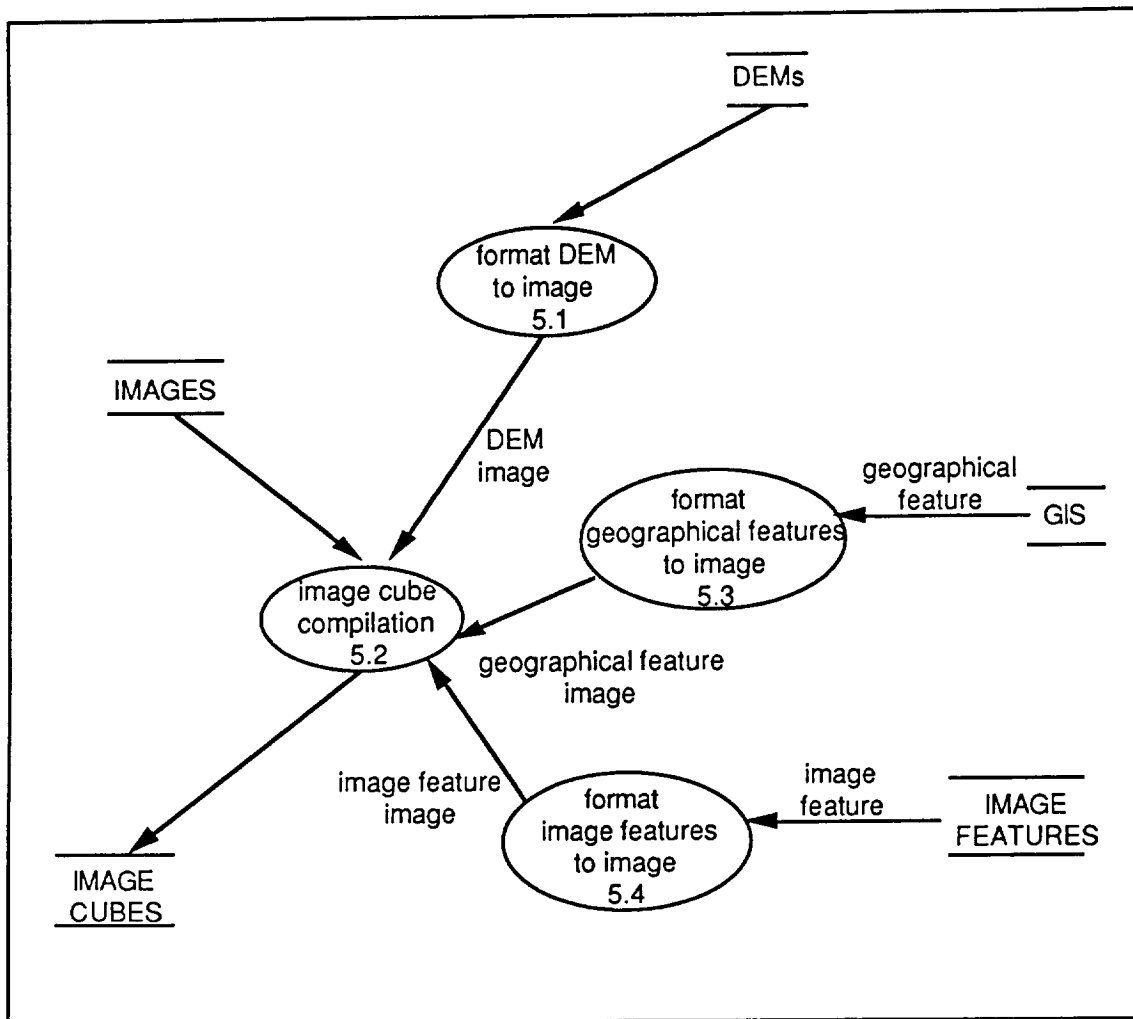


Figure 6.2: IMAGE CUBE GENERATION (Bubble 5 in Figure 6.1)



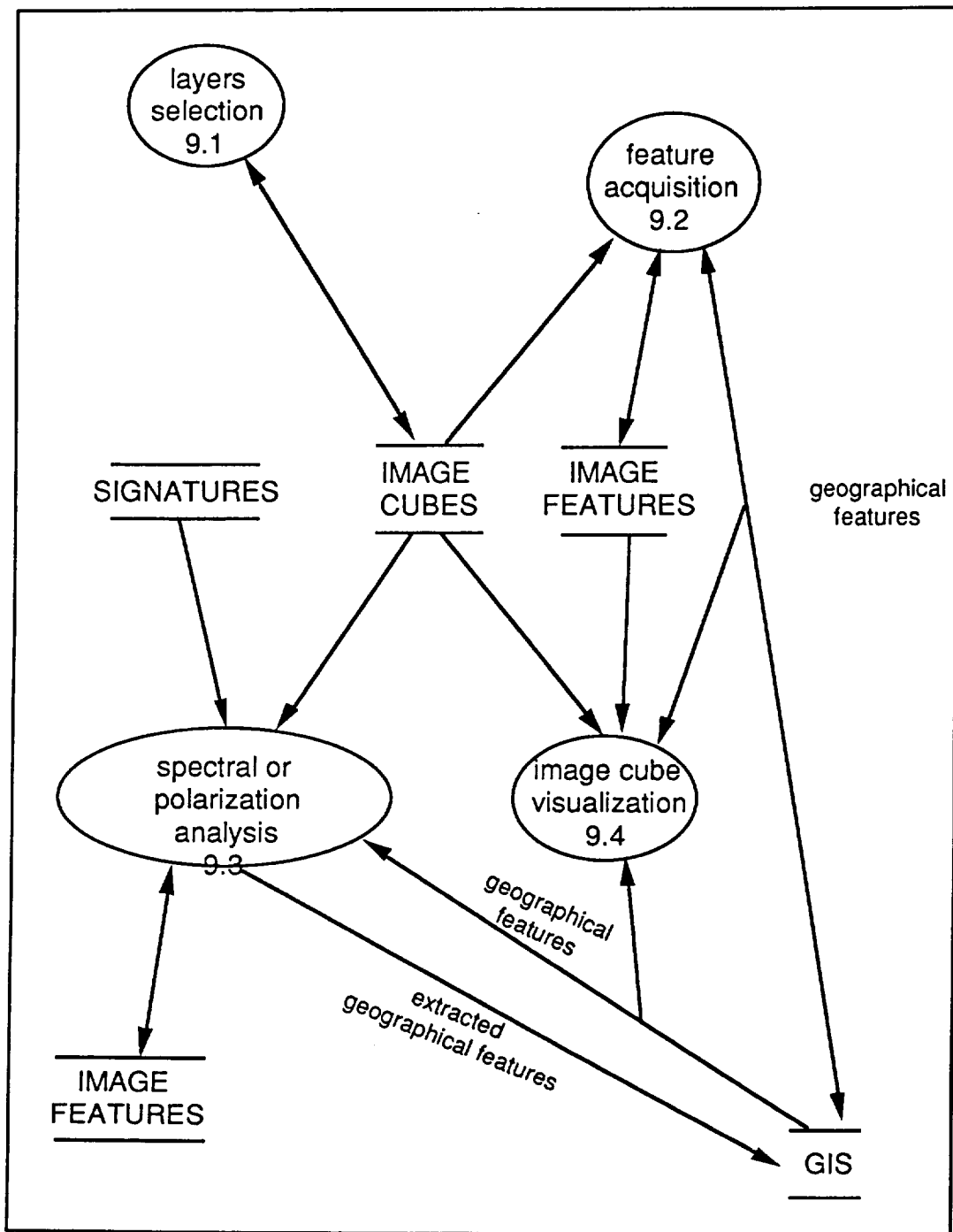


Figure 6.3: IMAGE CUBE ANALYSIS (Bubble 9 in Figure 6.1)

—

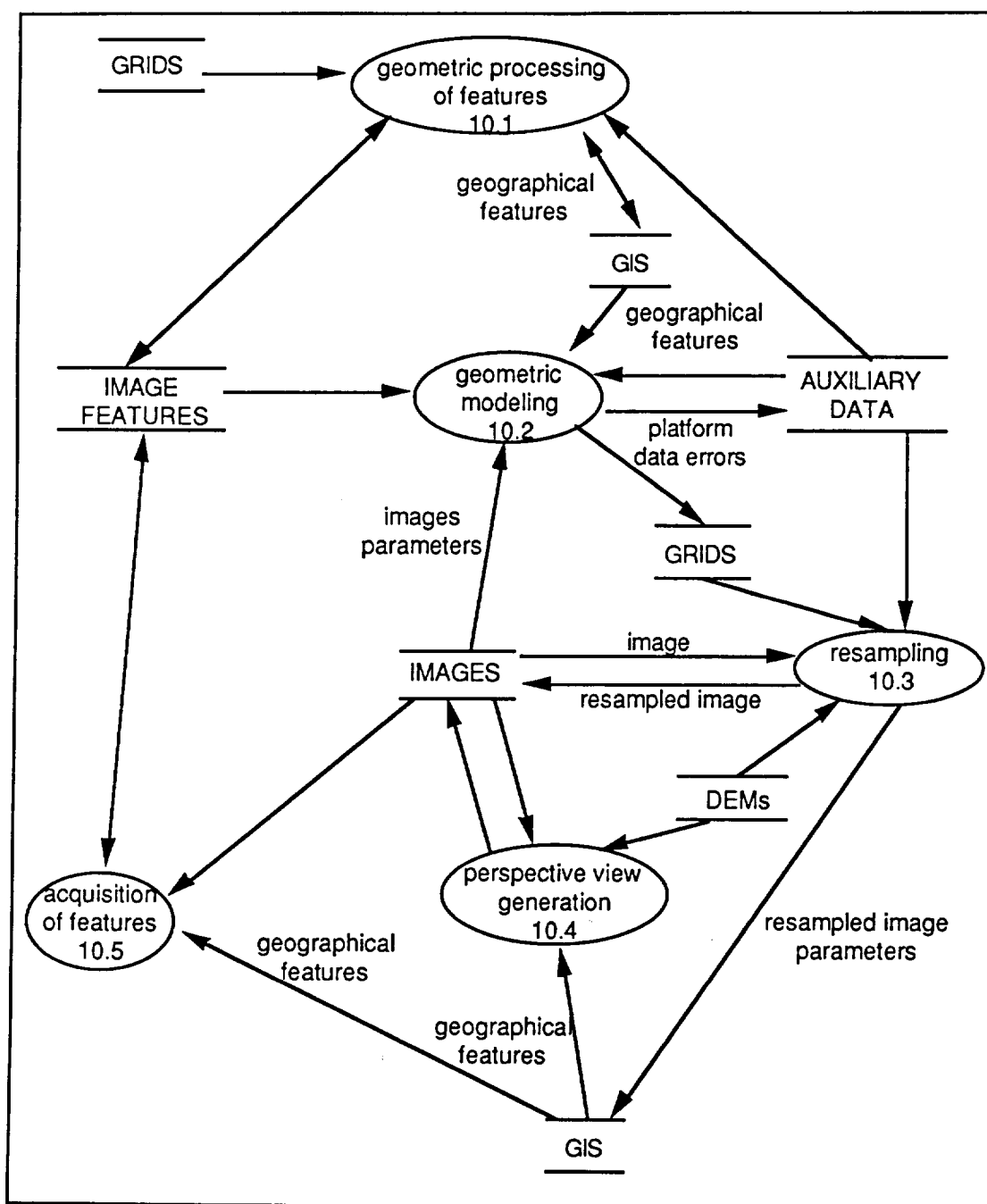


Figure 6.4: GEOMETRIC PROCESSING (Bubble 10 in Figure 6.1)

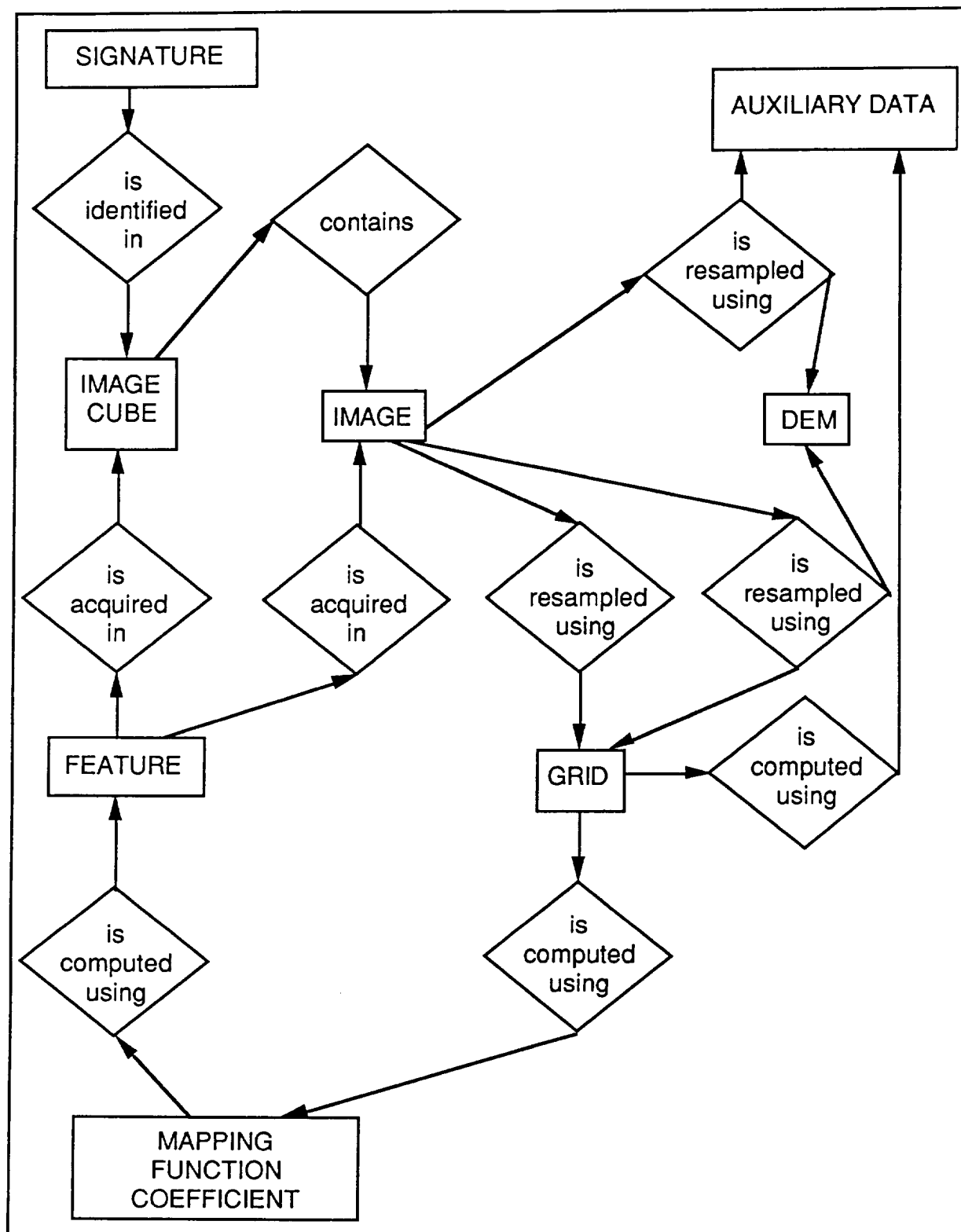


Figure 6.5: ENTITY-RELATIONSHIP DIAGRAM

6.2 Data Dictionary

This data dictionary represents an overview of definitions of terms and representations in data files.

The following conventions apply:

{ } iteration

[] select one of the alternative choices

| separates alternative choices in the [] construct

* * comment

+ and

1. *Introduction*
 2. *Background*
 3. *Methodology*
 4. *Results*
 5. *Discussion*
 6. *Conclusion*
 7. *References*
 8. *Appendix*
 9. *Figure 1*
 10. *Figure 2*
 11. *Figure 3*
 12. *Figure 4*
 13. *Figure 5*
 14. *Figure 6*
 15. *Figure 7*
 16. *Figure 8*
 17. *Figure 9*
 18. *Figure 10*
 19. *Figure 11*
 20. *Figure 12*
 21. *Figure 13*
 22. *Figure 14*
 23. *Figure 15*
 24. *Figure 16*
 25. *Figure 17*
 26. *Figure 18*
 27. *Figure 19*
 28. *Figure 20*
 29. *Figure 21*
 30. *Figure 22*
 31. *Figure 23*
 32. *Figure 24*
 33. *Figure 25*
 34. *Figure 26*
 35. *Figure 27*
 36. *Figure 28*
 37. *Figure 29*
 38. *Figure 30*
 39. *Figure 31*
 40. *Figure 32*
 41. *Figure 33*
 42. *Figure 34*
 43. *Figure 35*
 44. *Figure 36*
 45. *Figure 37*
 46. *Figure 38*
 47. *Figure 39*
 48. *Figure 40*
 49. *Figure 41*
 50. *Figure 42*
 51. *Figure 43*
 52. *Figure 44*
 53. *Figure 45*
 54. *Figure 46*
 55. *Figure 47*
 56. *Figure 48*
 57. *Figure 49*
 58. *Figure 50*
 59. *Figure 51*
 60. *Figure 52*
 61. *Figure 53*
 62. *Figure 54*
 63. *Figure 55*
 64. *Figure 56*
 65. *Figure 57*
 66. *Figure 58*
 67. *Figure 59*
 68. *Figure 60*
 69. *Figure 61*
 70. *Figure 62*
 71. *Figure 63*
 72. *Figure 64*
 73. *Figure 65*
 74. *Figure 66*
 75. *Figure 67*
 76. *Figure 68*
 77. *Figure 69*
 78. *Figure 70*
 79. *Figure 71*
 80. *Figure 72*
 81. *Figure 73*
 82. *Figure 74*
 83. *Figure 75*
 84. *Figure 76*
 85. *Figure 77*
 86. *Figure 78*
 87. *Figure 79*
 88. *Figure 80*
 89. *Figure 81*
 90. *Figure 82*
 91. *Figure 83*
 92. *Figure 84*
 93. *Figure 85*
 94. *Figure 86*
 95. *Figure 87*
 96. *Figure 88*
 97. *Figure 89*
 98. *Figure 90*
 99. *Figure 91*
 100. *Figure 92*
 101. *Figure 93*
 102. *Figure 94*
 103. *Figure 95*
 104. *Figure 96*
 105. *Figure 97*
 106. *Figure 98*
 107. *Figure 99*
 108. *Figure 100*
 109. *Figure 101*
 110. *Figure 102*
 111. *Figure 103*
 112. *Figure 104*
 113. *Figure 105*
 114. *Figure 106*
 115. *Figure 107*
 116. *Figure 108*
 117. *Figure 109*
 118. *Figure 110*
 119. *Figure 111*
 120. *Figure 112*
 121. *Figure 113*
 122. *Figure 114*
 123. *Figure 115*
 124. *Figure 116*
 125. *Figure 117*
 126. *Figure 118*
 127. *Figure 119*
 128. *Figure 120*
 129. *Figure 121*
 130. *Figure 122*
 131. *Figure 123*
 132. *Figure 124*
 133. *Figure 125*
 134. *Figure 126*
 135. *Figure 127*
 136. *Figure 128*
 137. *Figure 129*
 138. *Figure 130*
 139. *Figure 131*
 140. *Figure 132*
 141. *Figure 133*
 142. *Figure 134*
 143. *Figure 135*
 144. *Figure 136*
 145. *Figure 137*
 146. *Figure 138*
 147. *Figure 139*
 148. *Figure 140*
 149. *Figure 141*
 150. *Figure 142*
 151. *Figure 143*
 152. *Figure 144*
 153. *Figure 145*
 154. *Figure 146*
 155. *Figure 147*
 156. *Figure 148*
 157. *Figure 149*
 158. *Figure 150*
 159. *Figure 151*
 160. *Figure 152*
 161. *Figure 153*
 162. *Figure 154*
 163. *Figure 155*
 164. *Figure 156*
 165. *Figure 157*
 166. *Figure 158*
 167. *Figure 159*
 168. *Figure 160*
 169. *Figure 161*
 170. *Figure 162*
 171. *Figure 163*
 172. *Figure 164*
 173. *Figure 165*
 174. *Figure 166*
 175. *Figure 167*
 176. *Figure 168*
 177. *Figure 169*
 178. *Figure 170*
 179. *Figure 171*
 180. *Figure 172*
 181. *Figure 173*
 182. *Figure 174*
 183. *Figure 175*
 184. *Figure 176*
 185. *Figure 177*
 186. *Figure 178*
 187. *Figure 179*
 188. *Figure 180*
 189. *Figure 181*
 190. *Figure 182*
 191. *Figure 183*
 192. *Figure 184*
 193. *Figure 185*
 194. *Figure 186*
 195. *Figure 187*
 196. *Figure 188*
 197. *Figure 189*
 198. *Figure 190*
 199. *Figure 191*
 200. *Figure 192*
 201. *Figure 193*
 202. *Figure 194*
 203. *Figure 195*
 204. *Figure 196*
 205. *Figure 197*
 206. *Figure 198*
 207. *Figure 199*
 208. *Figure 200*
 209. *Figure 201*
 210. *Figure 202*
 211. *Figure 203*
 212. *Figure 204*
 213. *Figure 205*
 214. *Figure 206*
 215. *Figure 207*
 216. *Figure 208*
 217. *Figure 209*

Type	Definition
ACQUISITION DATA	PLATFORM DATA + SENSOR DATA
ACQUISITION DATE	year + month + day + hour + minute + second
ACQUISITION PARAMETERS	PLATFORM PARAMETERS + SENSOR PARAMETERS
AUXILIARY DATA	platform name + number of sensors (ns) + 1 {sensor name + number of images (ni) + 1 {image names}ni} ns + ACQUISITION PARAMETERS + ACQUISITION DATA
CARTOGRAPHICAL ORIGIN	X + Y + height * X units: meter range: variable Y units: meter range: variable height units: meter range: 0 - 9000*
DATUM DEFINITION	ELLIPSOID PARAMETERS + translation + rotation + scale *translation units: meter range: 0 - 2000 rotation units: radian range: infinitesimal scale units: none range: 0 - 1.0e-5*
DEM	DEM PARAMETERS + DEM VALUES
DEMs	{DEM}
DEM ACQUISITION PARAMETERS	*where the DEM comes from for example: - digitized contour lines - stereo photogrammetry*
DEM IMAGE	* DEM formatted to an image*
DEM PARAMETERS	DEM ACQUISITION PARAMETERS + GEOGRAPHICAL PARAMETERS + number of nodes in X + number of nodes in Y + grid step in X + grid step in Y * grid step units: decimal degree range: variable *

Type	Definition
DEM VALUES	$1 \{1 \{Height\} N_x\} N_y$ * Height units: 10 centimeters range: 0 - 90000 *
ELLIPSOID PARAMETERS	semi major axis + semi minor axis * semi major axis units: meter range: 6378000-6379000 semi minor axis units: meter range: 6356000-6357000*
EXTRACTED GEOGRAPHICAL FEATURES	* geographical features acquired in an mage cube*
FEATURE	number of references (Nr) + 1 {FEATURE REFERENCE PARAMETERS + number of points in the feature(n) + 1 {(latitude + longitude + height) + (L + P))n + FEATURE TYPE} Nr + FEATURE NATURE * longitude units: decimal degree range: -180 - + 180 latitude units: decimal degree range: -90 - + 90 height units: meter range: 0 - 9000 L units: pixel range: 1 - variable P units: pixel range: 1 - variable *
FEATURES	{FEATURE}
FEATURE NATURE	feature name + feature code
FEATURE REFERENCE	REFERENCE NAME + SPATIAL PARAMETERS PARAMETERS
FEATURE TYPE	[point area line]
GEOGRAPHICAL FEATURES IMAGE	*geographical features formatted to an image*
GEOGRAPHICAL ORIGIN	latitude + longitude + height * for units and range see FEATURE *
GEOGRAPHICAL FEATURE	* FEATURE with feature reference as the ground *
GEOGRAPHICAL FEATURES	{GEOGRAPHICAL FEATURE}

—

Type	Definition																								
GEOGRAPHICAL PARAMETERS	DATUM DEFINITION + (PROJECTION PARAMETERS) + ([CARTOGRAPHICAL ORIGIN GEOGRAPHICAL ORIGIN])																								
GRID	GRID PARAMETERS + GRID VALUES																								
GRID PARAMETERS	grid reference name + value reference name + pixel size + grid step in X + grid step in Y + number of nodes in X + number of nodes in Y + SPATIAL PARAMETERS * grid step units: meter/degrees/pixels range: variable*																								
GRID VALUES	$[[1\{1\{X + Y\}N_x\}N_y]1\{1\{\text{latitude} + \text{longitude}\}N_x\}N_y]]$ $1\{1\{L + P + (dL/dH + dP/dH) + [(dX/dH + dY/dH)(d\text{latitude}/dH + d\text{longitude}/dH)]\}N_x\}N_y\}$ <p>* Nx number of nodes along the X axis Ny number of nodes along the Y axis</p> <table><tr><td>X</td><td>units:</td><td>meters</td></tr><tr><td></td><td>range:</td><td>variable</td></tr><tr><td>Y</td><td>units:</td><td>meter</td></tr><tr><td></td><td>range:</td><td>variable</td></tr><tr><td>L</td><td>units:</td><td>pixel</td></tr><tr><td></td><td>range:</td><td>1 - variable</td></tr><tr><td>P</td><td>units:</td><td>pixel</td></tr><tr><td></td><td>range:</td><td>1 - variable*</td></tr></table>	X	units:	meters		range:	variable	Y	units:	meter		range:	variable	L	units:	pixel		range:	1 - variable	P	units:	pixel		range:	1 - variable*
X	units:	meters																							
	range:	variable																							
Y	units:	meter																							
	range:	variable																							
L	units:	pixel																							
	range:	1 - variable																							
P	units:	pixel																							
	range:	1 - variable*																							
IMAGE	image name + IMAGE PARAMETERS + PROCESSING HISTORY + IMAGE REFERENCE NAME + IMAGE VALUES																								
IMAGES	{IMAGE}																								
IMAGE CUBE	{ (IMAGE) + (DEM IMAGE) + (GEOGRAPHICAL FEATURES IMAGE) + (IMAGE FEATURE IMAGE)}																								
IMAGE CUBES	{IMAGE CUBE}																								
IMAGE FEATURE	* feature acquired in an image *																								
IMAGE FEATURES	{ IMAGE FEATURE}																								
IMAGE FEATURE IMAGE	* features acquired in an image formatted to an image *																								

Type	Definition
IMAGE PARAMETERS	number of lines + number of pixels + pixel size + acquisition date + sensor
name + number of bands (n) + 1{band	name}n + SPATIAL PARAMETERS * pixel size units: meter range: variable*
IMAGE REFERENCE NAME	* reference where is the image*
IMAGE VALUES	1{1{1{INTENSITY}NP}NL}NB *NP number of pixels per line NL number of lines NB number of bands*
MAPPING FUNCTION COEFFICIENTS	* coefficients of the function chosen to represent the mapping between 2 manifestations of an object (image or ground)*
PLATFORM DATA	GEOGRAPHICAL PARAMETERS + number of position and velocities samples (Np) + 1{time + latitude + longitude + H + (VX + VY + VH)}Np + (number of attitude samples(Na) + 1{roll + pitch + yaw}Na) * VX units: meter/second range: variable VY units: meter/second range: variable VZ units: meter/second range: variable roll units: radian range: variable pitch units: radian range: variable yaw units: radian range: variable*
PLATFORM DATA ERRORS	GEOGRAPHICAL PARAMETERS + number of position and velocities error samples (Np) + 1{time + (dlatitude) + (dlongitude) + (dH) + (VX + VY + VH) }Np + (number of attitude errors samples(na) + 1{droll + dpitch + dyaw}Na)

— — — — —

Type	Definition
	* for units and range see PLATFORM DATA*
PLATFORM PARAMETERS	* platform specific parameters*
RADAR SIGNATURE	material name + {polarity + {frequency + backscattering coefficient}} * frequency units: GHz range: 0 - 20 backscattering coefficient units: dB range: 60*
PROCESSING HISTORY	number of processes (n) + 1 { process name+input name(s)+output name(s) }n
PROJECTION PARAMETERS	projection name + parameters specific to this projection
REFERENCE ELLIPSOID	* default ellipsoid for processing and storage*
REFERENCE NAME	[ground image name image cube name + image cube name]
RESAMPLED IMAGE	**
SENSOR DATA	image name + number of time samples (Nt) + 1{time, line number} Nt + sensor specific data
SENSOR PARAMETERS	* describes mainly the nominal geometry of the sensor*
SENSOR SPECIFIC DATA	* data describing the specific physical behavior of a particular sensor*
SIGNATURES	{signature}
SIGNATURE	[spectral signature radar signature]
SPATIAL PARAMETERS	[GEOGRAPHICAL PARAMETERS image origin]
SPECTRAL SIGNATURE	material name + {wavelength + reflectance} * wavelength units: micrometers range: 0.2 - 15 reflectance units: T.B.D. range: T.B.D. *

ATTACHMENT B

DATABASE ANALYSIS DOCUMENTS

- **Database Design**
- **Database Prototype**

1. The first step is to identify the problem or question that needs to be addressed. This involves understanding the context and the specific requirements of the task.

EOS WORKSTATION

DATABASE DESIGN

author: **M.MILLOT**

date: **09/03/90**



INTRODUCTION.	2
HIGH LEVEL DESIGN.	2
IMAGE.	2
DEM.	4
FEATURE.	4
RESAMPLING GRID.	5
MIDDLE LEVEL DESIGN.	5
IMAGES TABLES AND ASSOCIATES.	6
REMOTE SENSING IMAGE.	6
IMAGE BANDS.	7
GEOMETRICALLY RAW IMAGE.	8
GEOMETRICALLY CORRECTED IMAGE.	8
IMAGE GEOMETRICALLY CORRECTED TO ANOTHER IMAGE.	9
IMAGE GEOMETRICALLY CORRECTED TO MAP.	9
IMAGE GEOMETRICALLY CORRECTED TO MAP WITHOUT DEM.	10
ORTHO IMAGE.	10
IMAGE-GRID.	10
IMAGE FILES.	11
REFERENCE SYSTEM TABLES AND ASSOCIATES.	11
REFERENCE SYSTEM.	11
COORDINATE SYSTEM.	12
UNITS OF MEASURE FOR GROUND PLANIMETRIC COORDINATES.	13
SPHEROID.	13
DEM TABLES AND ASSOCIATES.	14
DEM.	14
DEM FILES.	15
FEATURE TABLES AND ASSOCIATES.	16
FEATURE.	16
FEATURE GROUND COORDINATES.	16
GROUND-FEATURE.	17
FEATURE IMAGE COORDINATES.	18
IMAGE-FEATURE.	18
GROUND-IMAGE.	19
RESAMPLING GRID TABLES AND ASSOCIATES.	19
RESAMPLING GRID.	19
IMAGE RESAMPLING GRID.	20
MAP RESAMPLING GRID.	20
MAP RESAMPLING GRID WITHOUT DEM.	20
ORTHO RESAMPLING GRID.	21
RESAMPLING GRID FILES.	21

CONCLUSION.	22
------------------	----

Introduction

This document represents the design of a database for the Eos workstation. The main purpose of this database is to allow the potential user to store or retrieve information about geometrically processed images, DEM and features acquired in the images. Another class of objects important from the point of view of geometric processing is the resampling grid. It contains all the geographical information of the future resampled image. The design methodology used is from the article "Relational Database Design Using An Object-Oriented Methodology" by Michael R. BLAHA, William J. PREMERLANI and James E. RUMBAUGH Communications of the ACM, April 1988, Vol 31 #4.

High level design

From the geometric processing perspective, I identified 4 classes:

- . Images with their different levels of geometric processing,
- . DEM. They are used to generate ortho images,
- . Features used mainly to estimate the mapping between an image and another reference system (image or ground),
- . Resampling grid. They contain the geographical information of the future resampled image and represents the existing links between an image and other reference systems.

Image (see figure 1)

The class image is an example of the generalization relationship (represented by triangles in figure 1). A remote sensing image may be registered to an image or to a map at different level of processing (see figure 1). Each level contains attributes specific to this level.

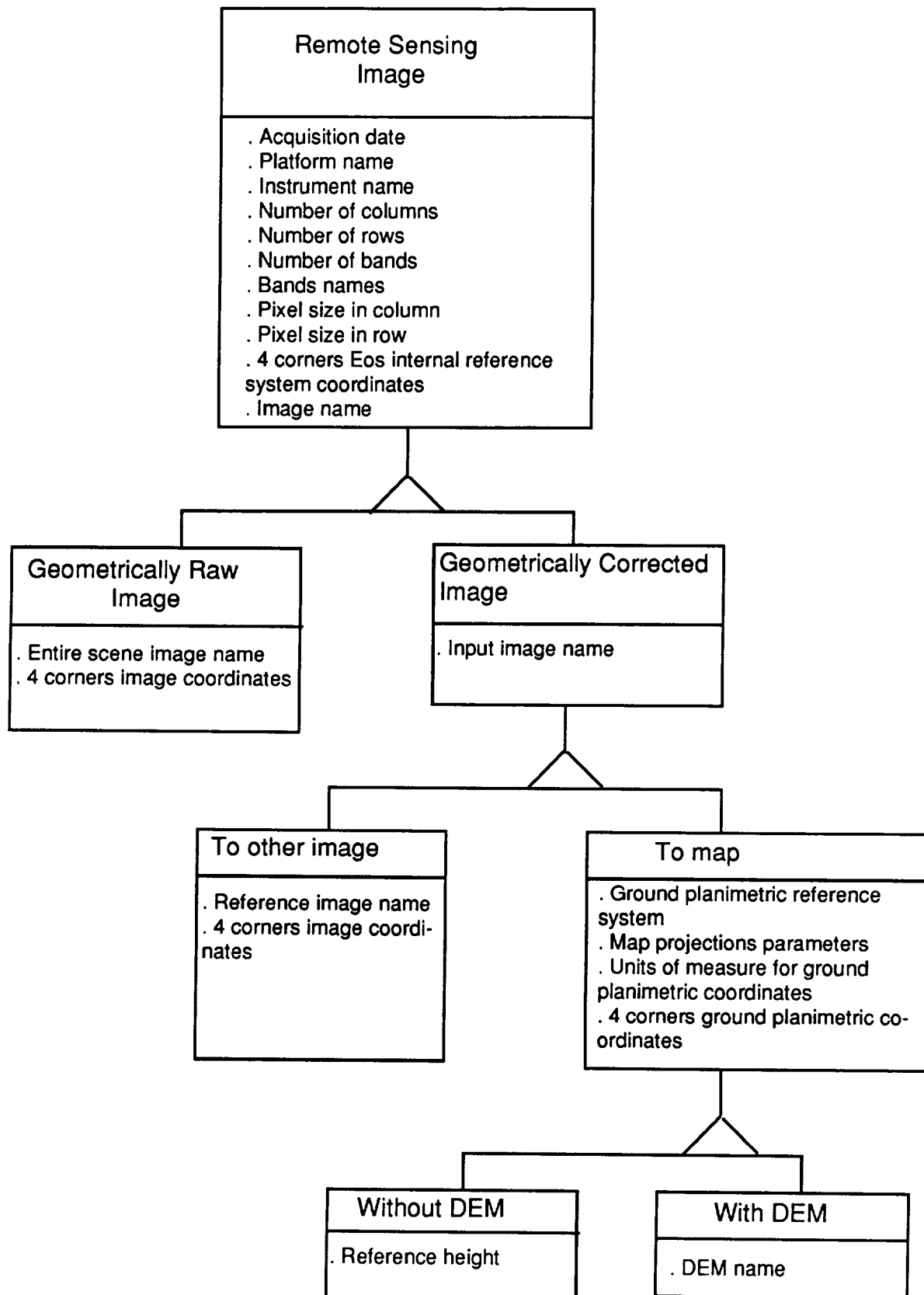


Figure 1 : Image



DEM (see figure 2)

Two attributes in the class DEM have to be described:

- . Input DEM name. This field will contain the name of the input DEM used to compute this DEM. The computation may be projection into another cartographic reference (the projection will always imply interpolation between the existing nodes) or editing of an existing noisy DEM.
- . 4 corners. As you can notice the 4 corners coordinates are represented into 2 different reference systems the Eos internal reference system and the DEM reference system. This redundancy is motivated by the desire to represent all DEMs in a common reference system.

DEM
<ul style="list-style-type: none">. Input DEM name. Ground planimetric reference system. Map projection parameters. Unit of measure for ground planimetric coordinates. Unit of measure for elevation coordinates. 4 corners Ground planimetric coordinates. Grid step in columns. Grid step in rows. Number of rows. Number of columns. DEM name. 4 corners Eos internal reference system coordinates

Figure 2: DEM

Features (see figure 3)

The feature class represents an example of aggregation (top to bottom arrows) and association (left to right line). This figure represents the following relationships:

- . a feature may have several sets of Ground coordinates,
- . a feature may have several sets of image coordinates,
- . ground coordinates may be computed using one or several sets of image coordinates.

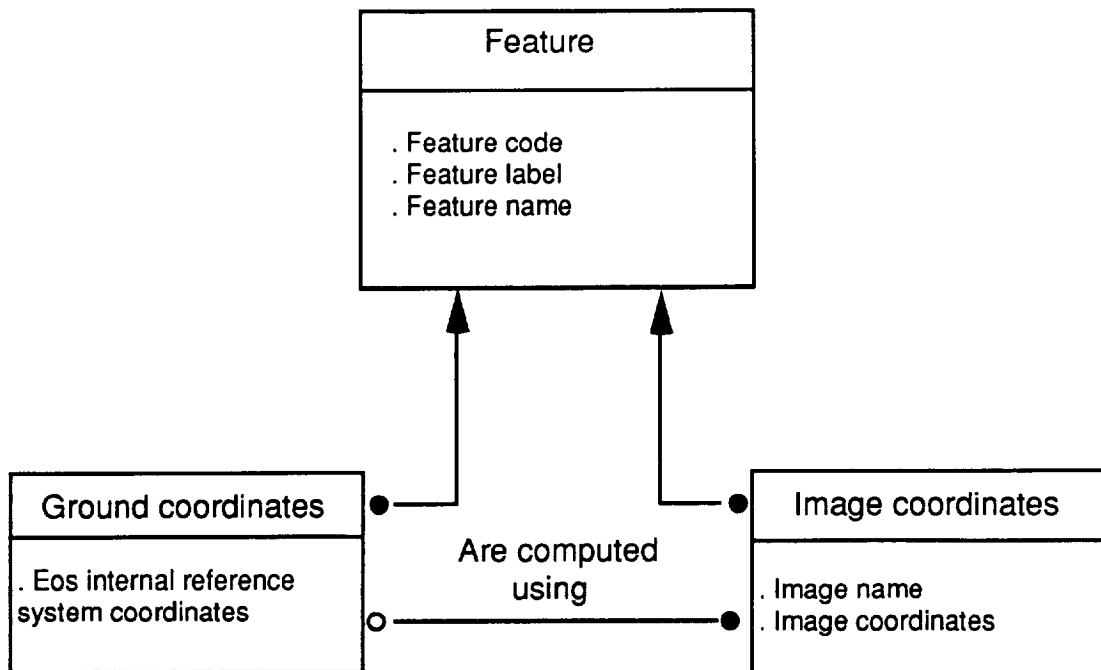


Figure 3: Feature

Resampling grid (see figure 4)

This class duplicates the generalization relationship encountered in the image class from the Geometrically Corrected Image subclass. They differ by the inclusion of the grid step attributes in the resampling grid class.

Middle level design

At this level are described the candidates tables and their corresponding candidates keys, independently of a particular RDBMS. The high level relationships are decomposed into tables satisfying if possible the third normal form. The following tables contain:

- . The attribute names.
- . If an attribute can be unknown or not applicable for a given row (the NULLS ? field)
- . An indication of the attribute domain.

It was chosen to represent each row in each table by an ID (unique number), providing a uniform mechanism for referencing all objects.

[illegible]

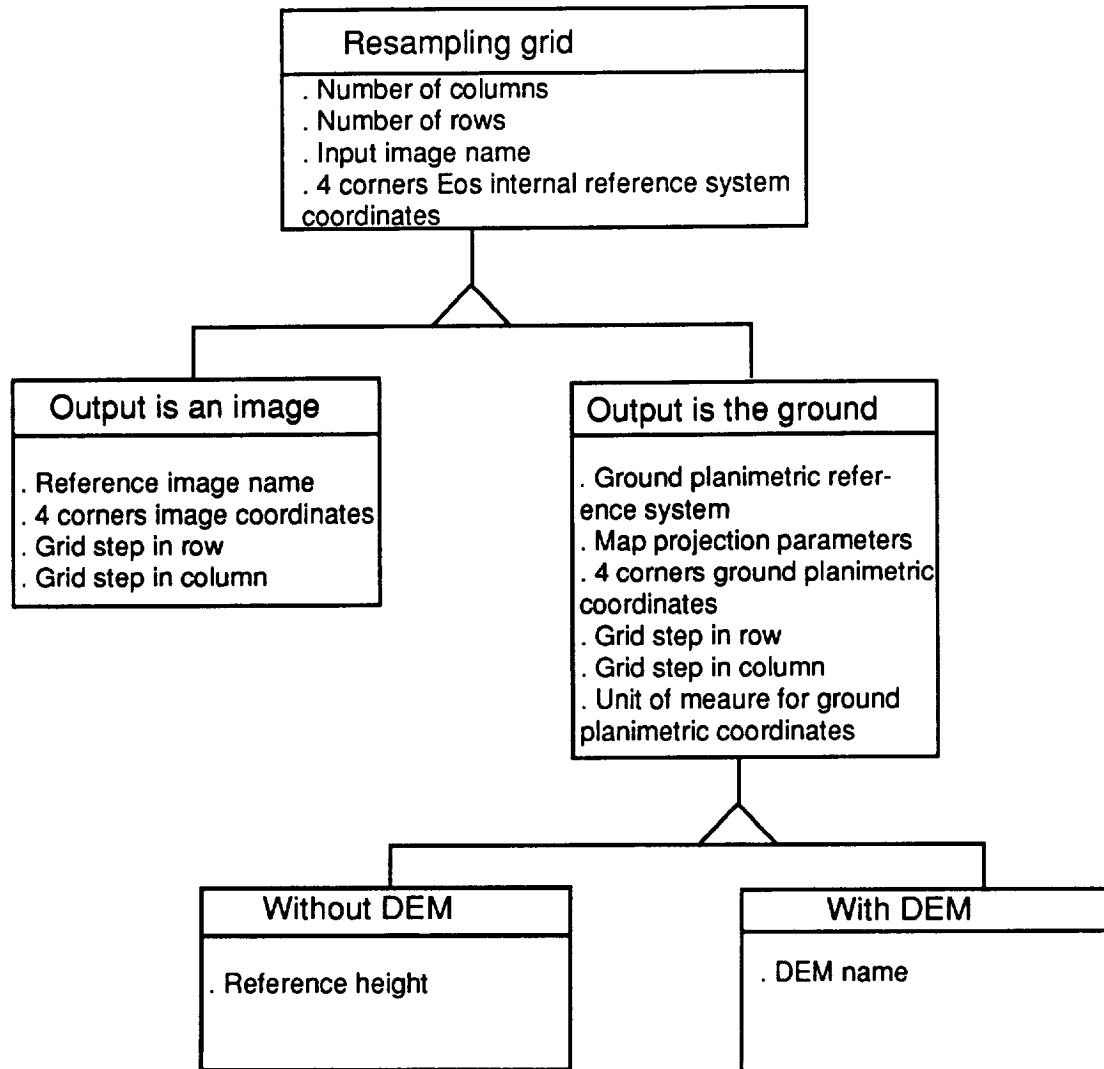


Figure 4 : Resampling grid

IMAGES TABLES AND ASSOCIATES

Remote sensing image (see figure 5)

The minimum amount of information required to store an image inside the database are:

- The instrument name. For remote sensing images the instrument name is always known.
- The size of the image, spatially and spectrally.
- The 4 corners coordinates. They may be only rough estimation.
- The image ID. A unique internal number attributed to an image.



REMOTE SENSING IMAGE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Acquisition date	Y	Date
Platform name	Y	Name
Instrument name	N	Name
Number of columns	N	Integer
Number of rows	N	Integer
Number of bands	N	Integer
Pixel size in column	Y	Length
Pixel size in row	Y	Length
4 corners EOS internal reference system coordinates	N	Distance
Image name	Y	Name
Image ID	N	ID

Candidate keys (Image ID)

Frequently accessed (Image ID) (Image name)

Figure 5

Image bands (see figure 6)

This table was created to satisfy the first normal form for the remote sensing image table. It contains the name and number of each spectral/polarization band included in one particular remote sensing image.

IMAGE BANDS

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID	N	ID
Band name	Y	Name
Band number	N	Integer

Candidate keys (Image ID)

Frequently accessed (Image ID)

Figure 6



Geometrically raw image (see figure 7)

This table allows the definition of a window inside a raw image and it is represented by :

- . The entire scene image ID. This information represents the link with the entire scene ancillary data.
- . The 4 corners image coordinates. This is the needed information to geometrically process a window in a raw image.

GEOMETRICALLY RAW IMAGE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID	N	ID
Entire scene image ID	Y	ID
4 corners image coordinates	N	Distance

Candidate keys (Image ID)

Frequently accessed (Image ID)

Figure 7

Geometrically corrected image (see figure 8)

A geometrically corrected image can be corrected outside the Eos workstation, that is why the input image ID may be unknown to the Eos database.

GEOMETRICALLY CORRECTED IMAGE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID	N	ID
Input image ID	Y	ID

Candidate keys (Image ID)

Frequently accessed (Image ID)

Figure 8



Image geometrically corrected to another image (see figure 9)

For this subclass we are assuming that the reference image is a known Eos workstation image. This may seem restrictive, but who would like an image registered to another image without having the reference image ?

IMAGE GEOMETRICALLY CORRECTED TO ANOTHER IMAGE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID	N	ID
Reference image ID	N	ID
4 corners image coordinates	N	Distance

Candidate keys (Image ID)

Frequently accessed (IMAGE ID)

Figure 9

Image geometrically corrected to map (see figure 10)

To be considered as registered to a map an image must have an associated reference system ID and the ground planimetric coordinates of its 4 corners must be known.

IMAGE GEOMETRICALLY CORRECTED TO MAP

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID	N	ID
Reference system ID	N	ID
4 corners ground planimetric coordinates	N	Distance

Candidate keys (Image ID)

Frequently accessed (Image ID)

Figure 10

Image geometrically corrected to map without DEM (see figure 11)

The reference height attribute is equivalent to the use of a flat DEM. It is the simplest approximation of the terrain. If the image was generated externally of the Eos workstation, this height information may not be recorded with the image.

IMAGE GEOMETRICALLY CORRECTED TO MAP WITHOUT DEM

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID Reference height	N Y	ID Height

Candidate keys (Image ID)

Frequently accessed (Image ID)

Figure 11

Ortho image (see figure 12)

An ortho image is an image registered to a map using a DEM. If the image was generated externally of the Eos workstation the DEM may not be known to the Eos database

ORTHO IMAGE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID DEM ID	N Y	ID ID

Candidate keys (Image ID)

Frequently accessed (Image ID)

Figure 12

Image-grid (see figure 13)

This table represents the association relationship between an image and the



resampling grid used to generate this image.

IMAGE-GRID

ATTRIBUTE NAME	NULLS ?	DOMAIN
Resampling grid ID	N	ID
Image ID	N	ID

Candidate keys (Image ID) (Resampling grid ID)

Frequently accessed (Image ID) (Resampling grid ID)

Figure 13

Image files (see figure 14)

The images are not physically stored in the database. This table link an image ID with the image file and also ancillary data file specific to this instrument and platform. When the image file name or image ancillary data file name attributes have a NULL value, it could mean that there is no ancillary data associated with this image or the files are not currently on disk.

IMAGE FILES

ATTRIBUTE NAME	NULLS ?	DOMAIN
Image ID	N	ID
Image file name	Y	Name
Image ancillary data file name	Y	Name

Candidate keys (Image ID)

Frequently accessed (Image ID)

Figure 14

REFERENCE SYSTEM TABLES AND ASSOCIATES

reference system (see figure 15)

The reference system tables describe the cartographic projection used and all

associated parameters. it follows USGS format used in their public domain conversion routines and in their DEM format.

REFERENCE SYSTEM

ATTRIBUTE NAME	NULLS ?	DOMAIN
Reference system ID	N	ID
Coordinate system ID	N	ID
Reference system parameters	Y	15 numbers
Units of measure for ground planimetric coordinates ID	N	ID
Spheroid ID	N	ID
Zone number	Y	Integer

Candidate keys (Reference system ID)

Frequently accessed (Reference system ID)

Figure 15

coordinate system (see figure 16)

The coordinate system contains the available projection names (UTM, Lambert,...). It was chosen to store them in a distinct table for the following reasons: a code is easier to manipulate than a name with all the possible variations, and this table gives a list of the available cartographic projections at one place inside the database.

COORDINATE SYSTEM

ATTRIBUTE NAME	NULLS ?	DOMAIN
Coordinate system ID	N	ID
Coordinate system name	N	Name

Candidate keys (Coordinate system ID)

Frequently accessed (Coordinate system ID)

Figure 16

Units of measure for ground planimetric coordinates (see figure 17)

This table was created for consistency with the coordinate system table approach and the units of measure for ground planimetric coordinates name attribute will contains information like: decimal degrees, feet, meters, ...

UNITS OF MEASURE FOR GROUND PLANIMETRIC COORDINATES

ATTRIBUTE NAME	NULLS ?	DOMAIN
Units of measure for ground planimetric coordinates ID	N	ID
Units of measure for ground planimetric coordinates name	N	Name

Candidate keys (Units of measure for ground planimetric coordinates ID)

Frequently accessed (Units of measure for ground planimetric coordinates ID)

Figure 17

Spheroid (see figure 18)

This table contains the definition of the different ellipsoids available in the Eos workstation. It was called spheroid because some projections requires a sphere as input.

SPHEROID

ATTRIBUTE NAME	NULLS ?	DOMAIN
Spheroid ID	N	ID
Spheroid name	N	N
Semi-major axis	N	Length
Semi-minor axis	N	Length

Candidate keys (Spheroid ID)

Frequently accessed (Spheroid ID)

Figure 18

DEM TABLES AND ASSOCIATES

DEM (see figure 19)

This table contains all the relevant information registered in a USGS DEM tape plus some additional information relevant to the Eos workstation like for example the 4 corners Eos internal reference coordinates. Note that our definition of a DEM is more restrictive than the USGS definition, we assume that a DEM is an array with a constant number of columns per row and is regularly spaced on the ground.

DEM

ATTRIBUTE NAME	NULLS ?	DOMAIN
DEM ID	N	ID
Input DEM ID	Y	ID
Reference system ID	N	ID
4 corners ground planimetric coordinates	N	Distance
Grid step in column	N	Length
Grid step in row	N	Length
DEM name	Y	Name
4 corners Eos internal reference system coordinates	N	Distance

Candidate keys (DEM ID)

Frequently accessed (DEM ID)

Figure 19

DEM files (see figure 20)

This table represents the link between a DEM in the database and the physical implementation of it.

DEM FILES

ATTRIBUTE NAME	NULLS ?	DOMAIN
DEM ID DEM file name	N Y	ID Name

Candidate keys (DEM ID)

Frequently accessed (DEM ID)

Figure 20

FEATURE TABLES AND ASSOCIATES

Feature (see figure 21)

The feature table contains generic information about this feature independently of any possible implementation or acquisition of this feature. At this time all attributes, except the ID, are optional, but this may change in the future

FEATURE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Feature ID Feature code Feature label Feature name	N Y Y Y	ID Code Name Name

Candidate keys (Feature ID)

Frequently accessed (Feature ID) (Feature code)

Figure 21

Feature ground coordinates (see figure 22)

This table contains the ground coordinates of a particular feature. Each feature ground coordinates ID may correspond to several points, that is why the key to this table is Feature ground coordinates ID plus point number in the feature.

[The page contains faint, illegible markings or bleed-through from another document.]

FEATURE GROUND COORDINATES

ATTRIBUTE NAME	NULLS ?	DOMAIN
Feature ground coordinates ID	N	ID
Point number	N	Integer
Eos internal reference system		
X coordinate	Y	Distance
Eos internal reference system		
Y coordinate	Y	Distance
Eos internal reference system		
Z coordinate	Y	Distance

Candidate keys (Feature ground coordinate ID, Point number)

Frequently accessed (Feature ground coordinate ID)

Figure 22

Ground-feature (see figure 23)

This table represents the one to many relationship between features and ground coordinates. One feature (one object on the ground) may have several ground coordinates, depending on the time or the accuracy of the measurements. The Status attribute means:

.computed. These ground coordinated were computed using image coordinates recorded in the database.

. measured. Not computed

GROUND-FEATURE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Feature ground coordinates ID	N	ID
Feature ID	N	ID
Status	N	"computed", "measured"

Candidate keys (Ground coordinates ID) (Ground coordinates ID Feature ID)

Frequently accessed (Feature ID) (Ground coordinates ID)

Figure 23

Feature image coordinates (see figure 24)

This table contains the image coordinates of a given feature and is equivalent to the feature ground coordinates.

FEATURE IMAGE COORDINATES

ATTRIBUTE NAME	NULLS ?	DOMAIN
Feature image coordinates ID	N	ID
Point number	N	Integer
Image ID	N	ID
Image coordinates	N	Distance

Candidate keys (image coordinates ID, Point number)

Frequently accessed (image coordinates ID) (Image ID)

Figure 24

Image-feature (see figure 25)

This table represents the one to many relationship between features and image coordinates. One feature (one object on the ground) may have several image coordinates.

IMAGE-FEATURE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Feature image coordinates ID	N	ID
Feature ID	N	ID
Status	N	"measured", "computed"

Candidate keys (Feature Image coordinates ID)

Frequently accessed (Image coordinates ID) (Feature ID)

Figure 25

Ground-image (see figure 26)

This table represents the one to many relationship between ground coordinates and image coordinates. For example, one set of ground coordinates may be computed by stereo intersection.

GROUND-IMAGE

ATTRIBUTE NAME	NULLS ?	DOMAIN
Feature ground coordinates ID	N	ID
Feature image coordinates ID	N	ID

Candidate keys (Ground coordinates ID) (Image coordinates ID)

Frequently accessed (Ground coordinates ID) (Image coordinates ID)

Figure 26

RESAMPLING GRID TABLES AND ASSOCIATES

resampling grid (see figure 27)

The resampling grid hierarchy follows the geometrically corrected hierarchy with the addition of the grid step attributes.

RESAMPLING GRID

ATTRIBUTE NAME	NULLS ?	DOMAIN
Resampling grid ID	N	ID
Number of columns	N	Integer
Number of rows	N	Integer
4 corners Eos internal reference system coordinates	N	Distance
Input Image ID	N	ID

Candidate keys (Resampling grid ID)

Frequently accessed (Resampling grid ID) (Input image ID)

Figure 27

1. *Chlorophyll a* (Chl *a*)
 2. *Chlorophyll b* (Chl *b*)
 3. *Chlorophyll c* (Chl *c*)
 4. *Chlorophyll d* (Chl *d*)
 5. *Chlorophyll e* (Chl *e*)
 6. *Chlorophyll f* (Chl *f*)
 7. *Chlorophyll g* (Chl *g*)
 8. *Chlorophyll h* (Chl *h*)
 9. *Chlorophyll i* (Chl *i*)
 10. *Chlorophyll j* (Chl *j*)
 11. *Chlorophyll k* (Chl *k*)
 12. *Chlorophyll l* (Chl *l*)
 13. *Chlorophyll m* (Chl *m*)
 14. *Chlorophyll n* (Chl *n*)
 15. *Chlorophyll o* (Chl *o*)
 16. *Chlorophyll p* (Chl *p*)
 17. *Chlorophyll q* (Chl *q*)
 18. *Chlorophyll r* (Chl *r*)
 19. *Chlorophyll s* (Chl *s*)
 20. *Chlorophyll t* (Chl *t*)
 21. *Chlorophyll u* (Chl *u*)
 22. *Chlorophyll v* (Chl *v*)
 23. *Chlorophyll w* (Chl *w*)
 24. *Chlorophyll x* (Chl *x*)
 25. *Chlorophyll y* (Chl *y*)
 26. *Chlorophyll z* (Chl *z*)
 27. *Chlorophyll aa* (Chl *aa*)
 28. *Chlorophyll ab* (Chl *ab*)
 29. *Chlorophyll ac* (Chl *ac*)
 30. *Chlorophyll ad* (Chl *ad*)
 31. *Chlorophyll ae* (Chl *ae*)
 32. *Chlorophyll af* (Chl *af*)
 33. *Chlorophyll ag* (Chl *ag*)
 34. *Chlorophyll ah* (Chl *ah*)
 35. *Chlorophyll ai* (Chl *ai*)
 36. *Chlorophyll aj* (Chl *aj*)
 37. *Chlorophyll ak* (Chl *ak*)
 38. *Chlorophyll al* (Chl *al*)
 39. *Chlorophyll am* (Chl *am*)
 40. *Chlorophyll an* (Chl *an*)
 41. *Chlorophyll ao* (Chl *ao*)
 42. *Chlorophyll ap* (Chl *ap*)
 43. *Chlorophyll aq* (Chl *aq*)
 44. *Chlorophyll ar* (Chl *ar*)
 45. *Chlorophyll as* (Chl *as*)
 46. *Chlorophyll at* (Chl *at*)
 47. *Chlorophyll au* (Chl *au*)
 48. *Chlorophyll av* (Chl *av*)
 49. *Chlorophyll aw* (Chl *aw*)
 50. *Chlorophyll ax* (Chl *ax*)
 51. *Chlorophyll ay* (Chl *ay*)
 52. *Chlorophyll az* (Chl *az*)
 53. *Chlorophyll aza* (Chl *aza*)
 54. *Chlorophyll abz* (Chl *abz*)
 55. *Chlorophyll acz* (Chl *acz*)
 56. *Chlorophyll adz* (Chl *adz*)
 57. *Chlorophyll aez* (Chl *aez*)
 58. *Chlorophyll afz* (Chl *afz*)
 59. *Chlorophyll agz* (Chl *agz*)
 60. *Chlorophyll ahz* (Chl *ahz*)
 61. *Chlorophyll aiz* (Chl *aiz*)
 62. *Chlorophyll ajz* (Chl *ajz*)
 63. *Chlorophyll akz* (Chl *akz*)
 64. *Chlorophyll alz* (Chl *alz*)
 65. *Chlorophyll amz* (Chl *amz*)
 66. *Chlorophyll anz* (Chl *anz*)
 67. *Chlorophyll aoz* (Chl *aoz*)
 68. *Chlorophyll apz* (Chl *apz*)
 69. *Chlorophyll aqz* (Chl *aqz*)
 70. *Chlorophyll arz* (Chl *arz*)
 71. *Chlorophyll asz* (Chl *asz*)
 72. *Chlorophyll atz* (Chl *atz*)
 73. *Chlorophyll auz* (Chl *auz*)
 74. *Chlorophyll avz* (Chl *avz*)
 75. *Chlorophyll awz* (Chl *awz*)
 76. *Chlorophyll axz* (Chl *axz*)
 77. *Chlorophyll ayz* (Chl *ayz*)
 78. *Chlorophyll ayz* (Chl *ayz*)
 79. *Chlorophyll azz* (Chl *azz*)
 80. *Chlorophyll azaa* (Chl *aza*)
 81. *Chlorophyll abz* (Chl *abz*)
 82. *Chlorophyll acz* (Chl *acz*)
 83. *Chlorophyll adz* (Chl *adz*)
 84. *Chlorophyll aez* (Chl *aez*)
 85. *Chlorophyll afz* (Chl *afz*)
 86. *Chlorophyll agz* (Chl *agz*)
 87. *Chlorophyll ahz* (Chl *ahz*)
 88. *Chlorophyll aiz* (Chl *aiz*)
 89. *Chlorophyll ajz* (Chl *ajz*)
 90. *Chlorophyll akz* (Chl *akz*)
 91. *Chlorophyll alz* (Chl *alz*)
 92. *Chlorophyll amz* (Chl *amz*)
 93. *Chlorophyll anz* (Chl *anz*)
 94. *Chlorophyll aoz* (Chl *aoz*)
 95. *Chlorophyll apz* (Chl *apz*)
 96. *Chlorophyll aqz* (Chl *aqz*)
 97. *Chlorophyll arz* (Chl *arz*)
 98. *Chlorophyll asz* (Chl *asz*)
 99. *Chlorophyll atz* (Chl *atz*)
 100. *Chlorophyll auz* (Chl *auz*)
 101. *Chlorophyll avz* (Chl *avz*)
 102. *Chlorophyll awz* (Chl *awz*)
 103. *Chlorophyll axz* (Chl *axz*)
 104. *Chlorophyll ayz* (Chl *ayz*)
 105. *Chlorophyll ayz* (Chl *ayz*)
 106. *Chlorophyll ayz* (Chl *ayz*)
 107. *Chlorophyll ayz* (Chl *ayz*)
 108. *Chlorophyll ayz* (Chl *ayz*)
 109. *Chlorophyll ayz* (Chl *ayz*)
 110. *Chlorophyll ayz* (Chl *ayz*)
 111. *Chlorophyll ayz* (Chl *ayz*)
 112. *Chlorophyll ayz* (Chl *ayz*)
 113. *Chlorophyll ayz* (Chl *ayz*)
 114. *Chlorophyll ayz* (Chl *ayz*)
 115. *Chlorophyll ayz* (Chl *ayz*)
 116. *Chlorophyll ayz* (Chl *ayz*)
 117. *Chlorophyll ayz* (Chl *ayz*)
 118. *Chlorophyll ayz* (Chl *ayz*)
 119. *Chlorophyll ayz* (Chl *ayz*)
 120. *Chlorophyll ayz* (Chl *ayz*)
 121. *Chlorophyll ayz* (Chl *ayz*)
 122. *Chlorophyll ayz* (Chl *ayz*)
 123. *Chlorophyll ayz* (Chl *ayz*)
 124. *Chlorophyll ayz* (Chl *ayz*)
 125. *Chlorophyll ayz* (Chl *ayz*)
 126. *Chlorophyll ayz* (Chl *ayz*)
 127. *Chlorophyll ayz* (Chl *ayz*)
 128. *Chlorophyll ayz* (Chl *ayz*)
 129. *Chlorophyll ayz* (Chl *ayz*)
 130. *Chlorophyll ayz* (Chl *ayz*)
 131. *Chlorophyll ayz* (Chl *ayz*)
 132. *Chlorophyll ayz* (Chl *ayz*

Image resampling grid (see figure 28)

IMAGE RESAMPLING GRID

ATTRIBUTE NAME	NULLS ?	DOMAIN
Resampling grid ID	N	ID
Reference image ID	N	ID
4 corners reference image coordinates	N	Distance
Grid step in row	N	Length
Grid step in column	N	Length

Candidate keys (Resampling grid ID)

Frequently accessed (Resampling grid ID) (Reference image ID)

Figure 28

Map resampling grid (see figure 29)

MAP RESAMPLING GRID

ATTRIBUTE NAME	NULLS ?	DOMAIN
Resampling grid ID	N	ID
Reference system ID	N	ID
4 corners ground planimetric coordinates	N	Distance
Grid step in row	N	Length
Grid step in column	N	Length

Candidate keys (Resampling grid ID)

Frequently accessed (Resampling grid ID)

Figure 29

Map resampling grid without DEM (see figure 30)

MAP RESAMPLING GRID WITHOUT DEM

ATTRIBUTE NAME	NULLS ?	DOMAIN
Resampling grid ID Reference height	N Y	ID Height

Candidate keys (Resampling grid ID)

Frequently accessed (Resampling grid ID)

Figure 30

Ortho resampling grid (see figure 31)

ORTHO RESAMPLING GRID

ATTRIBUTE NAME	NULLS ?	DOMAIN
Resampling grid ID DEM ID	N Y	ID ID

Candidate keys (Resampling grid ID)

Frequently accessed (Resampling grid ID)

Figure 31

Resampling grid files (see figure 32)

RESAMPLING GRID FILES

ATTRIBUTE NAME	NULLS ?	DOMAIN
Examples Resampling grid ID Resampling grid file name	Y Y Y	Names ID Name

Candidate keys (Resampling grid ID)

Frequently accessed (Resampling grid ID)

Figure 32



Conclusion

The above design can be summarized in several relation views. The first one (figure 33) shows a partial view of the database, without the generalization relationship, to emphasize the relationship between the principal classes. The second one shows the entire system.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

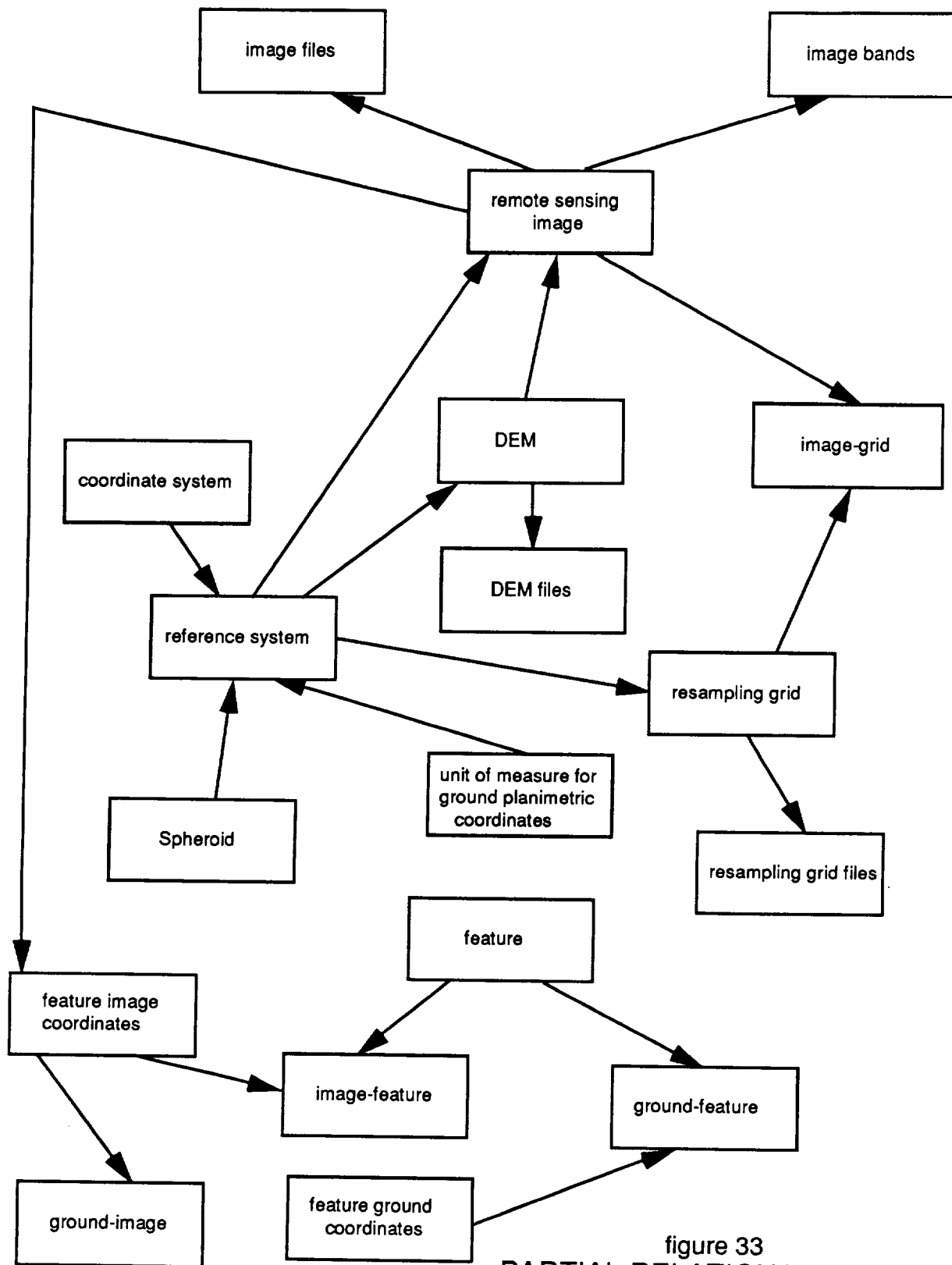
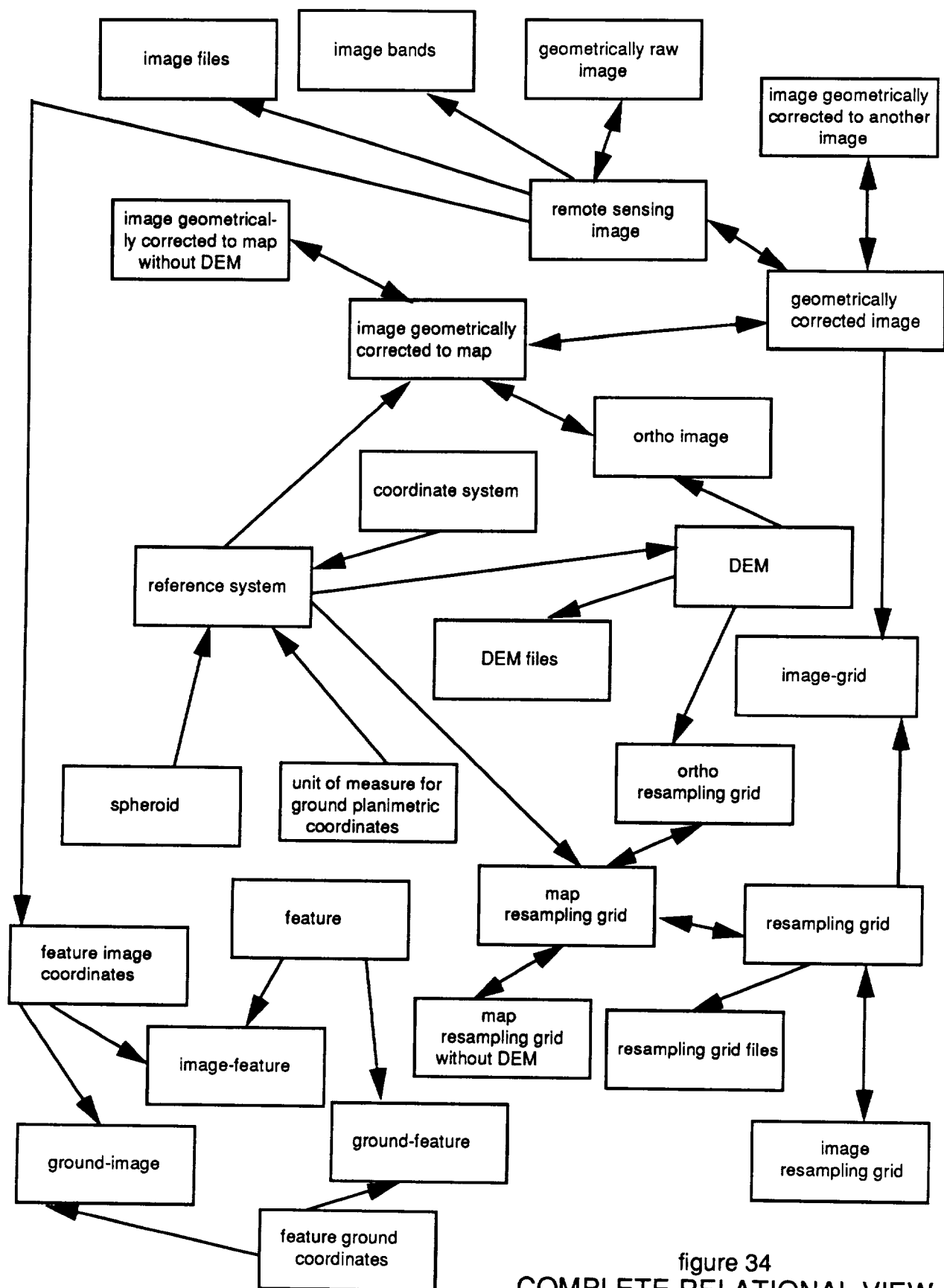


figure 33
PARTIAL RELATIONAL VIEW
(without generalization relationships)



EOS WORKSTATION

DATABASE PROTOTYPE

author: **M.MILLOT**

date: **09/14/90**

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

Introduction.....	2
Tables description.....	2
Retrieval examples.....	4
conclusion.....	7

1. **Introduction**
 2. **Background**
 3. **Methodology**
 4. **Results**
 5. **Discussion**
 6. **Conclusion**
 7. **References**
 8. **Appendix**
 9. **Figure 1**
 10. **Figure 2**
 11. **Figure 3**
 12. **Figure 4**
 13. **Figure 5**
 14. **Figure 6**
 15. **Figure 7**
 16. **Figure 8**
 17. **Figure 9**
 18. **Figure 10**
 19. **Figure 11**
 20. **Figure 12**
 21. **Figure 13**
 22. **Figure 14**
 23. **Figure 15**
 24. **Figure 16**
 25. **Figure 17**
 26. **Figure 18**
 27. **Figure 19**
 28. **Figure 20**
 29. **Figure 21**
 30. **Figure 22**
 31. **Figure 23**
 32. **Figure 24**
 33. **Figure 25**
 34. **Figure 26**
 35. **Figure 27**
 36. **Figure 28**
 37. **Figure 29**
 38. **Figure 30**
 39. **Figure 31**
 40. **Figure 32**
 41. **Figure 33**
 42. **Figure 34**
 43. **Figure 35**
 44. **Figure 36**
 45. **Figure 37**
 46. **Figure 38**
 47. **Figure 39**
 48. **Figure 40**
 49. **Figure 41**
 50. **Figure 42**
 51. **Figure 43**
 52. **Figure 44**
 53. **Figure 45**
 54. **Figure 46**
 55. **Figure 47**
 56. **Figure 48**
 57. **Figure 49**
 58. **Figure 50**
 59. **Figure 51**
 60. **Figure 52**
 61. **Figure 53**
 62. **Figure 54**
 63. **Figure 55**
 64. **Figure 56**
 65. **Figure 57**
 66. **Figure 58**
 67. **Figure 59**
 68. **Figure 60**
 69. **Figure 61**
 70. **Figure 62**
 71. **Figure 63**
 72. **Figure 64**
 73. **Figure 65**
 74. **Figure 66**
 75. **Figure 67**
 76. **Figure 68**
 77. **Figure 69**
 78. **Figure 70**
 79. **Figure 71**
 80. **Figure 72**
 81. **Figure 73**
 82. **Figure 74**
 83. **Figure 75**
 84. **Figure 76**
 85. **Figure 77**
 86. **Figure 78**
 87. **Figure 79**
 88. **Figure 80**
 89. **Figure 81**
 90. **Figure 82**
 91. **Figure 83**
 92. **Figure 84**
 93. **Figure 85**
 94. **Figure 86**
 95. **Figure 87**
 96. **Figure 88**
 97. **Figure 89**
 98. **Figure 90**
 99. **Figure 91**
 100. **Figure 92**
 101. **Figure 93**
 102. **Figure 94**
 103. **Figure 95**
 104. **Figure 96**
 105. **Figure 97**
 106. **Figure 98**
 107. **Figure 99**
 108. **Figure 100**
 109. **Figure 101**
 110. **Figure 102**
 111. **Figure 103**
 112. **Figure 104**
 113. **Figure 105**
 114. **Figure 106**
 115. **Figure 107**
 116. **Figure 108**
 117. **Figure 109**
 118. **Figure 110**
 119. **Figure 111**
 120. **Figure 112**
 121. **Figure 113**
 122. **Figure 114**
 123. **Figure 115**
 124. **Figure 116**
 125. **Figure 117**
 126. **Figure 118**
 127. **Figure 119**
 128. **Figure 120**
 129. **Figure 121**
 130. **Figure 122**
 131. **Figure 123**
 132. **Figure 124**
 133. **Figure 125**
 134. **Figure 126**
 135. **Figure 127**
 136. **Figure 128**
 137. **Figure 129**
 138. **Figure 130**
 139. **Figure 131**
 140. **Figure 132**
 141. **Figure 133**
 142. **Figure 134**
 143. **Figure 135**
 144. **Figure 136**
 145. **Figure 137**
 146. **Figure 138**
 147. **Figure 139**
 148. **Figure 140**
 149. **Figure 141**
 150. **Figure 142**
 151. **Figure 143**
 152. **Figure 144**
 153. **Figure 145**
 154. **Figure 146**
 155. **Figure 147**
 156. **Figure 148**
 157. **Figure 149**
 158. **Figure 150**
 159. **Figure 151**
 160. **Figure 152**
 161. **Figure 153**
 162. **Figure 154**
 163. **Figure 155**
 164. **Figure 156**
 165. **Figure 157**
 166. **Figure 158**
 167. **Figure 159**
 168. **Figure 160**
 169. **Figure 161**
 170. **Figure 162**
 171. **Figure 163**
 172. **Figure 164**
 173. **Figure 165**
 174. **Figure 166**
 175. **Figure 167**
 176. **Figure 168**
 177. **Figure 169**
 178. **Figure 170**
 179. **Figure 171**
 180. **Figure 172**
 181. **Figure 173**
 182. **Figure 174**
 183. **Figure 175**
 184. **Figure 176**
 185. **Figure 177**
 186. **Figure 178**
 187. **Figure 179**
 188. **Figure 180**
 189. **Figure 181**
 190. **Figure 182**
 191. **Figure 183**
 192. **Figure 184**
 193. **Figure 185**
 194. **Figure 186**
 195. **Figure 187**
 196. **Figure 188**
 197. **Figure 189**
 198. **Figure 190**
 199. **Figure 191**
 200. **Figure 192**
 201. **Figure 193**
 202. **Figure 194**
 203. **Figure 195**
 204. **Figure 196**
 205. **Figure 197**
 206. **Figure 198**
 207. **Figure 199**
 208. **Figure 200**
 209. **Figure 201**
 210. **Figure 202**
 211. **Figure 203**
 212. **Figure 204**
 213. **Figure 205**
 214. **Figure 206**
 215. **Figure 207**
 216. **Figure 208**
 217. **Figure 209**

Introduction

This document contains the results of prototyping the database described in the database design document dated 09/03/90. The prototyping was done on the SUN using SQL of INGRES.

Tables description

The prototype contains the following tables:

- . REMOTE SENSING IMAGE
- . IMAGE BANDS
- . FEATURE
- . FEATURE GROUND COORDINATES
- . GROUND-FEATURE
- . FEATURE IMAGE COORDINATES
- . IMAGE FEATURE
- . GROUND-IMAGE

They were chosen because they represent the most complex part of the design, the other relationships involving simple retrieval statements (like join or restrict). It was decided to work with a real set of data and the data were the ones collected and processed during the STAR1 calibration effort over the THREE HILLS mountains. More precisely were stored in the database: 8 images, the corresponding set of tie points and set of ground control points. The implementation was (In the following italic style are SQL statements):

- . REMOTE SENSING IMAGE table (8 rows):

```
create table image (instrument c10, acq_date c10, platform c10, nb_col smallint,
nb_row smallint, nb_band smallint, size_x float4, size_y float4, l1 smallint, p1
smallint, x1 float4, y1 float4, l2 smallint, p2 smallint, x2 float4, y2 float4, l3 small-
int, p3 smallint, x3 float4, y3 float4, l4 smallint, p4 smallint, x4 float4, y4 float4,
name c10, image_id integer)
```

```
copy table image (instrument=c0, acq_date=c0, platform=c0, nb_col=c0,
nb_row=c0, nb_band=c0, size_x=c0, size_y=c0, l1=c0, p1=c0, x1=c0, y1=c0,
l2=c0, p2=c0, x2=c0, y2=c0, l3=c0, p3=c0, x3=c0, y3=c0, l4=c0, p4=c0, x4=c0,
y4=c0, name=c0, image_id=c0)
from '/home/michel/ingres/original/image.table'
```

- . IMAGE BANDS table (8 rows). The image bands table was created from the image table in the following way:

```
create table bands (image_id integer, number smallint, name c10);
```



```
insert into bands(image_id)
select image_id from image;
```

```
update bands
set number =1;
```

. FEATURE table (2371 rows). the way the table was generated on the VAX created duplicated feature id, therefore the feature table loading was:

```
create table temp (feature_id integer, code c10, label c10, name c10);
```

```
copy table temp (feature_id=c0) from
'/home/michel/ingres/original/feature.table'
```

```
create table feature
as select distinct feature_id, code, label , name
from temp;
```

```
drop temp;
```

. FEATURE GROUND COORDINATES table (4097 rows):

```
create table fground (fground_id integer, pt_num smallint, x float4, y float4, z
float4);
```

```
copy table fground (fground_id=c6, pt_num=c4, x=c10, y=c10, z=c0nl)
from '/home/michel/ingres/original/fground.table';
```

```
modify fground to isam on fground_id;
```

. FEATURE IMAGE COORDINATES table (7826 rows):

```
create table fimage (fimage_id integer, pt_num smallint, image_id integer, p
float4, q float4);
```

```
copy table fimage (fimage_id=c6, pt_num=c4, image_id=c4, q=c10, p=c0nl)
from '/home/michel/ingres/original/fimage.table';
```

```
modify fimage to isam on fimage_id;
```

.GROUND-FEATURE table (4097 rows):

```
create table fea_gro (fground_id integer, feature_id integer, status c10);
```

```
copy table fea_gro (fground_id=c6, feature_id=c6, status=c0nl)
```

```

from '/home/michel/ingres/original/fea_fground.table';

modify fea_gro to isam on fground_id;

. IMAGE-FEATURE table (7826 rows):

create table fea_ima (fimage_id integer, feature_id integer, status c10);

copy table fea_ima (fimage_id=c6, feature_id=c6, status=c0nl)
from '/home/michel/ingres/original/fea_fimage.table';

modify fea_ima to isam on fimage_id;

. GROUND-IMAGE table (7826 rows):

create table gro_ima (fground_id integer, fimage_id integer);

copy table gro_ima (fground_id integer, fimage_id integer)
from '/home/michel/ingres/original/fground_fimage.table';

modify gro_ima to isam on fimage_id;

```

Transactions examples

In order to evaluate the correctness of the database design the following transactions examples were written and their results compared with the original data on the VAX:

. Extract the image coordinates of features acquired in a pair of images (it seems complex but the complexity is mainly caused by the fact that one tie point has 2 sets of image coordinates in the middle image and the only way to make the difference is by using the corresponding ground coordinates):

```

select i1.p, i1.q, i2.p, i2.q, i1.pt_num, f1.feature_id
from fimage i1, fimage i2, fea_ima f1, fea_ima f2, gro_ima g1, gro_ima g2
where i1.fimage_id in
    (select f.fimage_id from fimage f, image i
     where f.image_id = i.image_id
     and i.name='LW12_1A')
and i2.fimage_id in
    (select f.fimage_id from fimage f, image i
     where f.image_id = i.image_id
     and i.name='LW13_1A')
and f1.fimage_id = i1.fimage_id
and f2.fimage_id=i2.fimage_id
and f1.feature_id = f2.feature_id
and g1.fimage_id = i1.fimage_id

```



```
and g2.fimage_id = i2.fimage_id  
and g1.fground_id = g2.fground_id;
```

. Extract the computed and measured ground coordinates of Ground Control Points acquired in a pair of images (the temporary table was created because INGRES do not allow more than 10 tables in one statement and do not permit view creation on such statement (too many tables)):

```
create table temp as  
select g1.fground_id, f1.feature_id  
from fimage i1, fimage i2, fea_ima f1, fea_ima f2, gro_ima g1, gro_ima g2  
where i1.fimage_id in  
      (select f.fimage_id from fimage f, image i  
       where f.image_id = i.image_id  
       and i.name='LW12_1A')  
and i2.fimage_id in  
      (select f.fimage_id from fimage f, image i  
       where f.image_id = i.image_id  
       and i.name='LW13_1A')  
and f1.fimage_id = i1.fimage_id  
and f2.fimage_id=i2.fimage_id  
and f1.feature_id = f2.feature_id  
and g1.fimage_id = i1.fimage_id  
and g2.fimage_id = i2.fimage_id  
and g1.fground_id = g2.fground_id;
```

```
select g1.x, g1.y, g1.z, g2.x, g2.y, g2.z, g1.pt_num, t.feature_id  
from temp t, fea_gro f, fground g1, fground g2  
where t.feature_id = f.feature_id  
and f.status = 'MEASURED'  
and f.fground_id = g1.fground_id  
and t.fground_id = g2.fground_id
```

```
drop temp;
```

. Compute mean and rms of the differences between measured and computed coordinates of Ground Control Points acquired in a pair of images:

replace in the previous transaction:

```
select g1.x, g1.y, g1.z, g2.x, g2.y, g2.z, g1.pt_num, t.feature_id
```

by:

```
select mean_x = avg(g1.x-g2.x), mean_y = avg(g1.y-g2.y),  
mean_z = avg(g1.z-g2.z),
```



```

rms_x = sqrt(avg((g1.x-g2.x)**2)-avg(g1.x-g2.x)**2),
rms_y = sqrt(avg((g1.y-g2.y)**2)-avg(g1.y-g2.y)**2),
rms_z = sqrt(avg((g1.z-g2.z)**2)-avg(g1.z-g2.z)**2),
number = count(f.feature_id)

```

. Extract the 2 sets of ground coordinates of tie points acquired in a triplet of images.

```

create table temp1 as
select g1.fground_id, f1.feature_id
from fimage i1, fimage i2, fea_ima f1, fea_ima f2, gro_ima g1, gro_ima g2
where i1.fimage_id in
    (select f.fimage_id from fimage f, image i
     where f.image_id = i.image_id
     and i.name='LW12_1A')
and i2.fimage_id in
    (select f.fimage_id from fimage f, image i
     where f.image_id = i.image_id
     and i.name='LW13_1A')
and f1.fimage_id = i1.fimage_id
and f2.fimage_id=i2.fimage_id
and f1.feature_id = f2.feature_id
and g1.fimage_id = i1.fimage_id
and g2.fimage_id = i2.fimage_id
and g1.fground_id = g2.fground_id;

```

```

create table temp2 as
select g1.fground_id, f1.feature_id
from fimage i1, fimage i2, fea_ima f1, fea_ima f2, gro_ima g1, gro_ima g2
where i1.fimage_id in
    (select f.fimage_id from fimage f, image i
     where f.image_id = i.image_id
     and i.name='LW13_1A')
and i2.fimage_id in
    (select f.fimage_id from fimage f, image i
     where f.image_id = i.image_id
     and i.name='LW14_1A')
and f1.fimage_id = i1.fimage_id
and f2.fimage_id=i2.fimage_id
and f1.feature_id = f2.feature_id
and g1.fimage_id = i1.fimage_id
and g2.fimage_id = i2.fimage_id
and g1.fground_id = g2.fground_id;

```

```

select g1.x, g1.y, g1.z, g2.x, g2.y, g2.z, g1.pt_num, t1.feature_id
from temp1 t1, temp2 t2, fground g1, fground g2

```



```
where t1.feature_id = t2.feature_id  
and t1.fground_id = g1.fground_id  
and t2.fground_id = g2.fground_id;
```

```
drop temp1, temp2;
```

. Compute the mean and rms of the differences between the 2 sets of ground coordinates of tie points acquired in a triplet of images.

replace in the previous transaction:

```
select g1.x, g1.y, g1.z, g2.x, g2.y, g2.z, g1.pt_num, t.feature_id
```

by:

```
select mean_x = avg(g1.x-g2.x), mean_y = avg(g1.y-g2.y),  
mean_z = avg(g1.z-g2.z),  
rms_x = sqrt(avg((g1.x-g2.x)**2)-avg(g1.x-g2.x)**2),  
rms_y = sqrt(avg((g1.y-g2.y)**2)-avg(g1.y-g2.y)**2),  
rms_z = sqrt(avg((g1.z-g2.z)**2)-avg(g1.z-g2.z)**2),  
number = count(f.feature_id)
```

Conclusion

The proposed design was partially prototyped with success. Therefore it is proposed to use this design on flat files for the EOS workstation functions. One of the remaining question is: are we going to write the access to the database files inside each executable module or outside using a preprocessor and a post-processor? . This question is important for the light table and means: is SSCS going to adopt EOS files definition ?. My personal conclusion is it would be better to adopt the pre/post processor idea therefore minimizing the dependency with this temporary flat files organization.

ATTACHMENT C

**USER'S MANUAL
OF THE
SOFTCOPY LIGHT TABLE SOFTWARE "LT"**



Vexcel Corporation

User's Reference
for
LT - Vexcel's Software Light Table

Author: Matt Jackson

Last Revision: Friday, November 1, 1991

1. Introduction.	1
2. Stereo Viewing.	1
3. Getting Started.	2
4. Image Display.	6
5. Feature Acquisition.	8
6. Menus.	12
7. Errors.	15
8. Limitations	15
9. System Administrator's Guide.	16
10. Troubleshooting.	21
11. Shortcuts.	22
12. Customizing LT	23

1. Introduction

LT is an interactive program which provides basic functions for displaying stereo image pairs and measuring the parallax between features on those images. Image display functions include panning, zooming (in predefined increments), brightness/contrast adjustments, and changing the x- and y-parallax of the images. Parallax measurements are accomplished by overlaying feature marks on the images and writing the image space coordinates of the feature marks to ASCII files. Six different kinds of feature marks can be created; four are single point types, and two are stereo polylines. Feature marks can be selected, deleted, and selectively displayed or blanked.

This document explains how to operate LT, and should be read by anyone who will use LT to display images and edit feature marks. All system administration information such as file formats, configuration information, environment variables, and directory structure is contained in the **System Administrator's Guide** chapter.

2. Stereo Viewing

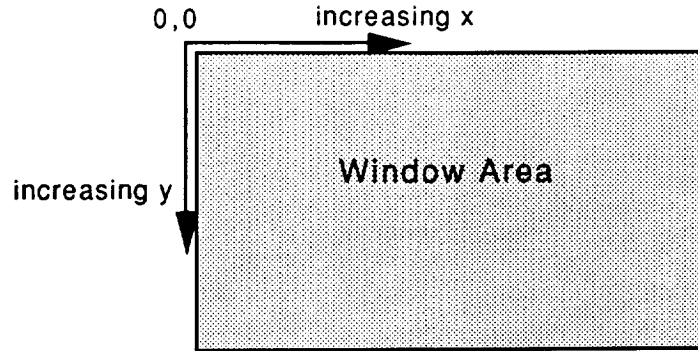
2.1. Concepts

The stereo impression you get from a system such as a Stardent computer with a Tektronix stereo monitor and stereo glasses is due to your left and right eyes seeing different images. The computer does this by displaying the two images alternately while the *LCD shutter* in front of the screen polarizes the light from the screen in two different ways in sync with the image display. The lenses on the stereo glasses are polarized so that each only lets the light from one of the images reach the eye.

If the image pairs have been constructed so that each is the view that one of your eyes would have if you were looking at an object from some distance, the sensation of looking at that object from that distance is recreated by the stereo viewing system.

If we could freeze and superimpose the images on your left and right retinae as you focus on a distant object, there would be two overlapping images – left and right perspective views – having what physiologists call *retinal disparity*, which is the horizontal distance between corresponding left and right retinal image points. Retinal disparity is what causes the sensation of depth. The distance between corresponding points on the left- and right-eye images displayed by a stereo viewing system is called *parallax*. Changing the parallax of the displayed images changes the retinal disparity of the images you see, which changes your perception of depth.

Actually, there are two kinds of parallax, since the images can be shifted left/right or up/down. The parallax caused by left/right shifts of the images is called *x-parallax* because LT uses the X Windows coordinate system for all graphics calculations:



Shifts to the left or right of one image relative to the other will be calculated by LT as changes in the x position of one of the images, while shifts up or down will be calculated as changes in y.

Since your eyes never use y-parallax to focus on an object in normal viewing, images which have y-parallax between features that you wish to focus on are very disturbing. They "feel" wrong, and the stereo impression is destroyed. Viewing images with even small amounts of y-parallax gives most people a headache rather quickly.

2.2. The Dot Cursor

Most graphics programs have a *cursor*, which is a small image on the screen that follows the movements of a *pointing device* (typically a mouse, trackball, thumb wheels, joystick, etc). It is usually used to point at some place on the screen in order to create a new object or to select the object at that location. LT's cursor is a small green dot which follows the motions of the mouse on the mouse pad. It lets you point at locations and objects on the screen in the usual way, but it also provides a height reference because it always appears in both eyes with no x- or y-parallax. When you change the x-parallax of the images, you are changing the apparent distance between you and the image features you are focusing on. Since the cursor's parallax doesn't change, it appears to dive into or float above the images depending on whether you have increased or decreased the x-parallax between the images. When you are acquiring points with LT, the object is to remove y-parallax and adjust the x-parallax so that the dot appears to rest on the surface of the terrain you are viewing. See **Parallax Adjustments** for an explanation of how to change the image parallax.

3. Getting Started

3.1. Using the Mouse

In addition to moving the dot cursor on the screen, the mouse is used to create and select feature marks, choose functions from menus, and change the values of various controls. The buttons on the mouse are used to *click on* or to *drag* screen objects. To click on something, place the cursor on the object, then press and release the mouse button. To drag, place the cursor on the object, then hold down the mouse button

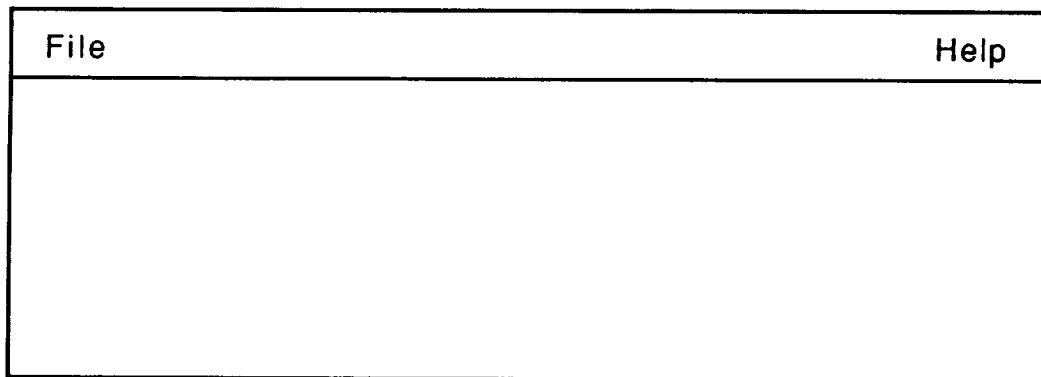
and move the mouse while keeping the button held down. The object moves as you move the mouse. Sometimes *double-clicking* can be used as a shortcut. A double-click is just two clicks in rapid succession.

In LT, use the left mouse button to pull down menus, choose functions from menus, change the values of controls, and create feature marks. Use the middle mouse button to change the x-parallax of the images, and the right mouse button to pop up the feature type menu.

3.2. Starting Up

When you first log in, the mono screen will be visible. This screen does not support stereo, and running LT here will not do you very much good. To change to the stereo screen, hold down the **Alt** key, then press **F2**.

To start LT, make sure that the program is in your search path (if you don't know what this means, get your system administrator to help), then type `lt`. After a short pause, the *initial window* will appear:

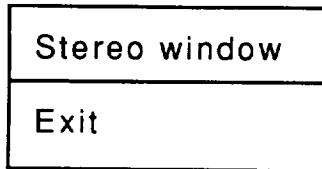


The area across the top which contains the **File** and **Help** labels is called the *menu bar*, because each label has a *pulldown menu* connected to it. Holding down the left mouse button on one of the *menu buttons* makes a menu appear. Make a choice from the menu by moving the mouse down through the choices (still holding down the left button), until the selection you want is highlighted. Releasing the mouse button while your selection is highlighted performs the highlighted action. If you move the pointer out of the menu area and release the button, no action is performed and the menu disappears.

3.3. Initial Window Menus

3.3.1. File Menu

The **File** menu looks like this:



3.3.1.1. Stereo Window

Selecting **Stereo Window** from the **File** menu either pops up a dialog box prompting you for the name of your *project directory* or it pops up a *file selection box* which you can use to select the images you want to display. See **The Project Directory** and **Selecting Images** for more information.

3.3.1.2. Exit

Selecting **Exit** from the **File** menu exits LT.

3.3.2. Help Menu

The **Help** menu looks like this:



3.3.2.1. About LT

Selecting **About LT** from the **Help** menu displays a copyright notice and some basic information about LT.

3.4. The Project Directory

LT looks for images and feature data files in a *project directory*, which needs to be set up in the proper way before you can do any real work. LT finds this directory by checking a UNIX environment variable, which you can set in the startup files that are executed when you log in (see the **System Administrator's Guide** for information about how to do this). If this variable is not defined, when you select **Stereo window** from the initial window's **File** menu, a dialog box will appear, prompting you for the name of a directory to use as the project directory. Clicking on the **Cancel** button in this dialog box quits LT. If you click on **OK** after typing in a directory name, LT checks that the directory exists and that you have permission to create files in it, then displays the *file selection box*.

3.4.1. Selecting Images

When LT has located the project directory, it displays a *file selection box*, which consists (from top to bottom) of a *directory mask*, a list of images, a *selection area*, and a row of buttons.

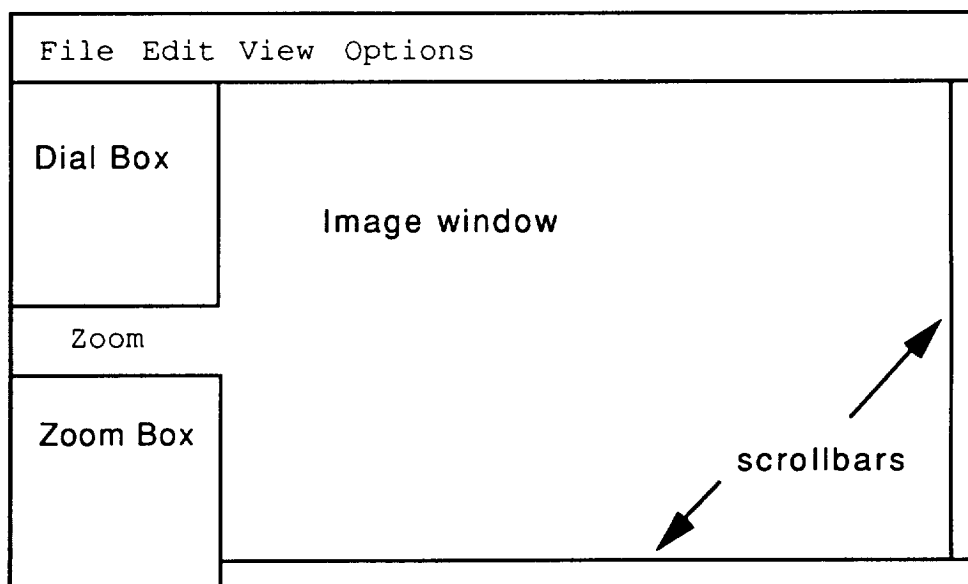
There are two ways to select a file name from the list. Clicking on a name in the list highlights it, and it is copied to the selection area. Then you confirm your choice by clicking on the **OK** button. Double-clicking on a file name in the list is a shortcut that does the same thing as clicking on the file name, then on the **OK** button. If you make a mistake, you can click on the **Cancel** button to restart the selection process.

You probably don't need to worry about the directory mask, since LT has already used the project directory to figure out which image files are available. The file name list contains the names of all image files in the project directory. If the list is long enough, there will be scrollbars on the right and/or bottom of the list so you can scroll through the choices. See **Panning** for an explanation of how to use the scrollbars.

At first, the label above the selection area reads **Left-eye image:**. After you have selected a file name for the left-eye image, this changes to read **Right-eye image:**. When you have selected images for both the left and right eyes, the file selection box disappears, and LT begins to read image data from files in the project directory. Then it displays the stereo window, reads feature data, and displays the images and controls. Depending on the size of the images, how much feature data there is, and how fully loaded your system is, this may take as long as a minute.

3.4.2. The Stereo Window

Here is a diagram of the stereo window:





The menu bar contains buttons that are connected to pulldown menus that you can access in the same way as the ones on the initial window. The dial box contains software dials that correspond to the knobs on the knob box. You can change the values of the dials with the mouse or by moving the corresponding knob on the knob box. Dragging the dial indicator changes the value of the dial in the same way as moving the knob, though you have less control when using the mouse. You can also click at some location on the perimeter of the dial, and the indicator will move to that location.

These dials let you change the amount of x- and y-parallax between the images and adjust their brightness and contrast.

The zoom box contains buttons that let you change to a new zoom level. These are called "radio" buttons because exactly one of them is always active, so they work like the buttons on a car radio.

The image window contains the images. Whenever you move the cursor into the image window, it changes to the dot cursor.

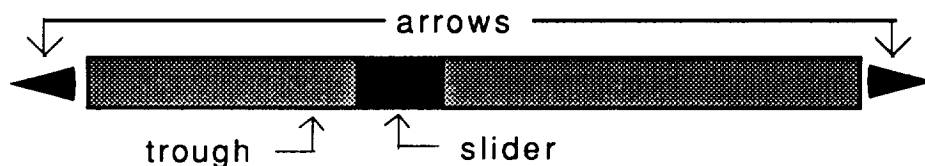
The scrollbars let you pan around the images without changing the parallax between them.

4. Image Display

This section describes how to use the controls in the stereo window to pan around the images, change the parallax between them, zoom in or out, and change the brightness and contrast of the images.

4.1. Panning

You can pan around the images (without changing the parallax between them) using the scrollbars on the right and bottom of the image window. A scrollbar looks like this:



The size of the slider relative to the size of the trough tells you how much of the image is currently visible. If you zoom out or change the size of the window, the slider size will change as well. The position of the slider in the trough gives you an idea of which part of the image you are looking at.

There are three ways to use the scrollbar to pan around the images:

- Dragging a scrollbar slider with the left mouse button causes the slider (and the

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

images) to follow the motion of the mouse until you release the button.

- Clicking the left mouse button in the trough moves the slider towards the place where you clicked. The images move one screen width.
- Clicking the left mouse button on one of the arrows at either end of the scrollbar moves the slider towards that end of the scrollbar. The images move one quarter screen width.

As you pan, you are changing the position of the window over the images, not moving the images underneath the window. This is why moving the slider on the horizontal scrollbar to the right causes the images to move to the left.

4.2. Adjusting the Parallax

You can change the parallax between the images using the upper four dials in the dial box. The uppermost two dials move the left- and right-eye images in x, while the next two down move the images in y. By changing the position of one image in the window while keeping the other fixed, you are changing the parallax between the images. If you want to keep track of the position of the images or the amount of parallax, the label underneath the dial you are changing will change to reflect the new position of the image. The upper-left two dials show the image space coordinates of the upper-left corner of the window on the left-eye image, while the upper-right two dials give the image space coordinates of the upper-left corner of the window on the right-eye image.

You can also change the x-parallax of the images (thus changing the apparent height of the dot cursor) by holding down the middle mouse button and moving the mouse towards or away from the screen. This drives the dot cursor into the terrain or pulls it back away from it.

Note: if moving the mouse towards the screen makes the dot float higher above the terrain, change the **Depth Reversal** switch on the front of the stereo monitor. See **Troubleshooting** for more information.

4.3. Zooming

The buttons in the zoom box let you change how far away from you the image appears to be. Zoom can also be defined in terms of the number of image pixels that each screen pixel represents. The buttons in the zoom box are labeled $1:n$, where n is the number of image pixels represented by one screen pixel. At a zoom level of $1:1$, the images are as close to you as they can get. At other zoom levels, you can see more of the image, but in less detail. Zooming out is useful if you want to see which parts of the image have feature marks on them, or to find the region where two images overlap. When you zoom, the point at the center of the image window stays at the center.

4.4. Brightness/Contrast Adjustments

The left dial in the third row of the dial box changes the brightness of the images. The range of values is -1 to 1, with 0 being the default. Moving the dial in the clockwise direction increases the brightness value. With a brightness value of -1, everything appears black; at 1, everything appears white.

The right dial in the third row changes the contrast of the images. The range of values is 0 to 1, with a default of 0.5. Moving the dial in the clockwise direction increases the contrast, making bright pixels brighter and dark pixels darker. A contrast value of 0 gives all pixels the same value. Whether this value is bright or dark is determined by the value of the brightness dial.

5. Feature Acquisition

Feature acquisition is the process of overlaying simple geometric entities onto a stereo pair of images. LT currently supports point and polyline feature classes. Here is a list of the features that LT supports, and a brief description of their use:

- LT uses margin points to automatically remove y-parallax at the cursor position, and establish the work region for the stereo pair.
- Tie points appear in 3 or more images (2 or more stereo pairs). The SSCS radar-grammetric software uses them to accurately position one flightpath relative to another.
- Grid points are acquired on a square grid, and are used by other parts of SSCS to perform stereo intersection.
- Random points allow you make parallax measurements without any additional semantics.
- Breaklines indicate areas where the elevation changes sharply.
- Drainlines indicate lines of monotonically changing elevation, such as rivers.

The following sections explain how to create and edit the features that you can overlay on the images.

5.1. Margin Points

If you display a pair of images with no associated margin points, the images come up zoomed all the way out, and at zero x- and y-parallax so that you can find the region where they overlap. The area where the images overlap is the only region where you can see stereo, and is called the *work region*.

To acquire margin points, select **Margin Points** from the **Acquire** submenu of the **Edit** menu. Each time you click the left mouse button in the image window, LT

will place a light blue X at the place where you clicked. Be careful to completely clear the y-parallax at each point (using the vertical position dials – see **Panning**). The more accurate you are when acquiring margin points, the better the automatic parallax removal will work.

Usually margin points are acquired in two columns as close to the vertical edges of the work region as possible. Once you have acquired margin points, LT uses them to remove y-parallax at the cursor position by locating the four closest margin points, and interpolating between their y-parallaxes to arrive a guess as to what the y-parallax at the cursor position should be. Once you have acquired four or more margin points, the automatic parallax removal will start working (unless you are still acquiring margin points).

If you display a pair of images that have margin points associated with them, the images will come up zoomed all the way in, and the window will be centered over the work region.

5.2. Breaklines and Drainlines

To acquire breaklines or drainlines, choose the appropriate selection from the **Acquire** submenu of the **Edit** menu. The first time you click the left mouse button in the image window, a small X (purple for breaklines, blue for drainlines) appears at the place where you clicked. Every time after that, when you press the button, LT draws a white ghost line between the last point on the line and the cursor position. When you release the button, the line changes color (to purple for breaklines, blue for drainlines), and LT places an X at the cursor position. To start a new line, select **Breaklines** or **Drainlines** again. If you start a new line, then decide that you want to add a point to some other line, hold down the **Shift** key, then click the left mouse button on a point in the line that you want to add to. The line will flash white briefly. Any points that you acquire after that will be added to the line that flashed.

5.3. Grid Points

Because there are likely to be a lot of grid points and they need to be acquired very rapidly, LT handles them differently from other types of feature marks. To acquire grid points, there must be at least four margin points defined. This is because LT calculates the size and placement of the grid so that it covers the work region.

To acquire grid points, select **Grid Points** from the **Acquire** submenu of the **Edit** menu. When you do this, LT finds the upper-leftmost visible grid intersection that doesn't contain an acquired or skipped point. It moves the dot cursor to that location and draws a ghost box around the grid intersection. You can acquire a point by clicking anywhere inside the box or by typing a. You can skip the point by clicking outside the box or by typing s. In either case, the ghost box disappears,

and LT moves the dot cursor to the next point. When you have acquired or skipped all of the grid intersections in the current window, LT displays a dialog box asking you if you want to go the next window. Click on the **Cancel** button to make the dialog box disappear and put LT into Select mode (see **Editing**). If you click on the **OK** button, LT finds the upper-leftmost grid intersection anywhere on the images that is unacquired and unskipped. It positions the window so that the grid intersection is visible, then moves the dot cursor to that point and draws the ghost box. When you have acquired or skipped the last grid point, LT displays a dialog box saying that all grid points have been acquired.

5.4. Tie Points

Tie points are acquired on both images simultaneously in the same way as margin points are (except that you choose **Tie Points** from the **Acquire** submenu). However, tie points are the only feature type that will be displayed when they are only defined in one image. Typically, you will want to “balance” the tie points that were only defined in one image (because they were acquired on both images in some other stereo pair) by specifying their location in the other image of the stereo pair you are working on. You can do this by selecting **Balance tie points** from the **Edit** menu. LT will find the first visible tie point that is only defined in one image, move the dot cursor to that location, undraw the point, and lock the image that the tie point is already defined in. Moving the mouse (without any buttons pressed down) towards or away from the screen changes the x-parallax between the images by moving the image that the tie point is *not* defined in. This ensures that the position of the tie point does not change in the image where it was already defined. Clicking the left mouse button acquires the point, and LT finds the next visible unbalanced point. If the next point is not in the window, LT displays a dialog box saying that the point is off screen and asking if you want to go there anyway. Click on the **Cancel** button to make the dialog box disappear and put LT into Select mode (see **Editing**). If you click on the **OK** button, LT finds the next unbalanced point anywhere on the images, centers the window over it, undraws it, and locks the opposite image. When all tie points have been balanced, LT displays a dialog box saying that all the points have been balanced.

5.5. Random Points

Random points are acquired on both images simultaneously in the same way as tie points and margin points are, but random points are not used by LT internally, and have no extra semantics associated with them. They are typically used to make simple parallax measurements for subsequent processing. Random points can be selected and edited in the same way as any other feature type.

5.6. Editing



Occasionally, you may make a mistake. Perhaps you didn't quite get the dot cursor on the ground, so that a tie point is floating in the air or is stuck into a mountain. Maybe you forgot to use the menus to start a new breakline, so that the point you acquired is appended to some unrelated line. LT has two editing functions that let you fix mistakes, **Delete** and **Undo**.

5.6.1. Selecting

Before you can edit feature marks you must *select* them. To select feature marks, choose **Select** from the **Acquire** submenu of the **Edit** menu, then click the left mouse button on the feature mark that you wish to select. You can also press down the left mouse button and drag a ghost box around a group of feature marks. When you release the button, everything inside the box is selected. Selected feature marks turn white to show that they are selected. If you select one or more feature marks, then decide that you want to select others as well, you can hold down the **Shift** key and select other feature marks. These will be added to the *selection set*. If you decide not to edit the selected feature marks, you can clear the selection set by clicking the left mouse button in an area of the window where there are no feature marks.

To select all feature marks of a set of types, choose **Select by type** from the **Edit** menu. LT displays a dialog box that contains a list of all feature types, a **Cancel** button, and an **OK** button. You can choose the feature types to select by clicking on the rectangles next to the desired feature types so that they appear to go into the screen. If you click on the **OK** button, the dialog box disappears and all members of all the feature types you chose will be selected. Click on the **Cancel** button to make the dialog box disappear without selecting anything.

5.6.2. Deleting

Choose **Delete** from the **Edit** menu to delete all feature marks in the selection set. When you delete feature marks that are single points (margin points, grid points, random points, and tie points), they disappear. When you delete points in a breakline or drainline, the line is redrawn so that the remaining points are connected.

Deleted feature marks are never written to files when you **Save**.

5.6.3. Undo

Another way to fix mistakes is to undo them. LT keeps track of the last 32 changes you made to feature marks, and lets you undo them in reverse order. The label on the **Undo** button tells you what will be undone. Here are the operations that can be undone:

- Creation of any type of feature mark
- Deletion of any set of feature marks
- Skipping a grid point
- Balancing a tie point

If LT cannot undo any operations, or you have undone all 32 operations, the **Undo** button reads: Can't Undo and is grayed out.

6. Menus

This section describes the menus available from the stereo window menu bar and their functions.

6.1. File Menu

The **File** menu looks like this:

Save
Quit

6.1.1. Save

Selecting **Save** from the **File** menu writes all feature data to disk, overwriting any existing files.

6.1.2. Quit

Selecting **Quit** from the **File** menu exits the stereo window. If you have made any changes to feature data since the last **Save** operation, a dialog box appears, asking you if you want to **Save** your changes before quitting.

6.2. Edit

The **Edit** menu looks like this:



Undo	ctrl u
Acquire	→
Select by type...	
Balance tie points	
Delete	ctrl d

6.2.1. Undo

Selecting **Undo** from the **Edit** menu undoes the most recent change you made to a feature mark. You can undo up to 32 operations in reverse order by repeatedly selecting **Undo**. The **Undo** button is labeled with the operation that will be undone. You can undo:

- Creation of any type of feature mark
- Deletion of any set of feature marks
- Skipping a grid point
- Balancing a tie point

6.2.2. Select by Type

Selecting **Select by Type** from the **Edit** menu pops up this dialog box:

<input type="checkbox"/> Margin points	<input type="checkbox"/> Drainlines
<input type="checkbox"/> Grid points	<input type="checkbox"/> Breaklines
<input type="checkbox"/> Tie points	<input type="checkbox"/> Random points
<input type="button" value="OK"/>	<input type="button" value="Cancel"/>

The squares next to the feature type names are Motif markers indicating that these are non-exclusive choices. You can choose a set of feature types to select. Click on the **OK** button to select all features belonging to the set of types you chose. Click on the **Cancel** button to make the dialog box disappear without selecting anything. If features of any of the chosen types are currently blanked, they become visible.

6.2.3. Acquire

Selecting **Acquire** from the **Edit** menu brings up a submenu containing a list of available feature types. Selecting one of these buttons enables creation of that feature type. See **Feature Acquisition** for more information. The top button of the submenu is labeled **Select**. Choosing this button enables selection of feature marks. See **Selecting** for more information.

6.2.4. Balance tie points

Selecting **Balance tie points** from the **Edit** menu lets you “balance” tie points that are only defined on one image as result of having been acquired on both images of some other stereo pair. See **Tie Points** for complete information.

6.2.5. Delete

Selecting **Delete** from the **Edit** menu deletes all features in the current selection set. Deleted feature marks are never written back to disk during **Save** operations.

6.3. View Menu

The **View** menu looks like this:

Move to (Left)
Move to (Right)
Display/Undisplay

6.3.1. Move To

Selecting **Move To** from the **View** menu displays dialog boxes prompting you for the new x and y coordinates of the upper-left corner of the window over the image. You can change the position of the left-eye image by selecting **Move To (Left)**, or change the position of the right-eye image by selecting **Move To (Right)**.


6.3.2. Display...

Selecting **Display...** from the **Edit** menu displays the non-exclusive feature type button box. You can pick the feature types you wish to display or blank (make invisible). Click on the **OK** button to blank or redisplay all members of the feature types you selected. Click on the **Cancel** button to make the dialog box disappear

without making any changes. Any feature types that are blanked are still written to disk when you **Save**.

6.4. Options Menu

Clicking on the **Options** button pulls down this menu:



Grid size

6.4.1. Grid Size

Selecting **Grid Size** from the **Options** menu displays a dialog box prompting you for the new grid size used in acquiring grid points. The value you enter is interpreted as the number of pixels in x and y between grid intersections. This button is grayed out if any grid points have already been acquired or loaded from a previous session.

7. Errors

Hopefully this will never happen, but occasionally LT may encounter an error condition. The number of possible errors is quite large, and includes internal bugs that cause LT to crash, trouble communicating with the X server, running out of memory, etc. Whenever an error occurs, LT will display a dialog box that describes the error and gives you the option to continue or quit. The best thing to do under these circumstances is to click on the **Continue** button and immediately try to **Save**. If this doesn't work, clicking on the **Quit** button will exit LT.

8. Limitations

- Although it is possible to display more than one stereo pair at a time in separate stereo windows, some feature acquisition functions are not guaranteed to work properly. These include grid points, margin points, and breaklines/drainlines.
- LT uses a lot of memory. When we loaded and displayed a database of 38,000 feature marks, LT and the X server together used 22.4 megabytes of memory.
- Because of its extreme memory usage, LT can cause problems for both its operator and everyone else on a fully loaded system. It works best if no one else is on the machine.



9. System Administrator's Guide

9.1. Hardware Requirements

LT runs on a Stardent 1500 or 3000 series computer with stereo support provided by a Tektronix stereo monitor or by the Stereographics CrystalEyes™ system. A mouse and keyboard are required, and a knob box is strongly recommended. Large amounts (≥ 32 MB) of RAM are highly recommended for performance reasons, though LT will run with only 24MB present.

9.2. Software Requirements

LT uses the X+ (X11R3 with extensions) graphics system provided by Stardent, and Motif 1.0.3. Stereo support is provided through the Stardent-specific Xd library. The Motif window manager (mwm) is not required, but is recommended because of LT's use of application-modal dialog boxes for some functions. The raster-to-Utiff image conversion program (ras2ut (1)) should be installed.

9.3. Startup Files

In order to use LT, the X server must be started with the `-pseudo` and `-stereo` options. `-stereo` tells the server to create two screens. The one that is visible at login time is the mono screen, and LT can only be run on the stereo screen. The user can change screens by typing `Alt/F2`. You should probably set up the user with `xterms` and a window manager on the stereo screen at login time. Here's one way to do it:

in `.login`:

```
xinit -- Xtitan -pseudo -stereo
```

in `.xinitrc`:

```
<normal stuff for the mono screen>
```

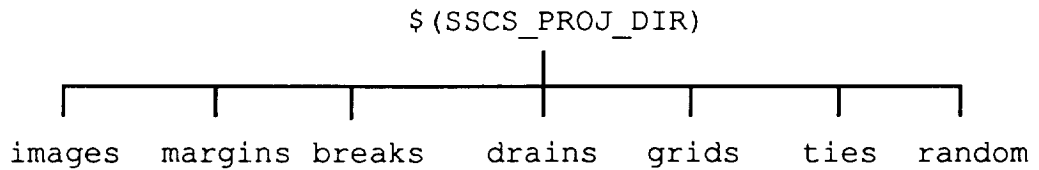
```
mwm -display unix:0.1 &
```

```
xterm -display unix:0.1 &
```

9.4. The Project Directory

Image files and feature data files are stored in a *project directory*, which LT locates by checking the `SSCS_PROJ_DIR` environment variable. If this variable is not set at startup, LT prompts the user for a directory name to use. This variable should be set in the startup files for any user that will be working on one project for a while. The project directory must have this structure:





The names of the subdirectories are X resources stored in the `Utchat` resource file.

9.5. Files

LT uses five different kinds of files:

- **Image files**

These contain the image data for the left- and right-eye images .

- **Feature Data files**

These contain the coordinates of feature marks that are overlaid on the images .

- **The resource file**

LT uses this file to configure the user interface (the size and placement of the panning controls, the text in dialog boxes, etc), to determine color and size of feature marks of various types, and to set up the action to be performed in response to mouse button clicks, key presses, etc.

- **The grid definition file**

This file defines parameters of the grid that need to be preserved across LT sessions such as the work region that the grid is defined on, and the grid spacing. If the grid definition file exists at window creation time, that definition of the grid is used to display all acquired grid points and acquire new ones. If the grid definition file does not exist, the grid is defined using the work region defined by the current set of margin points.

- **The grid skip file**

This file contains a list of the row/column coordinates of all grid points which were marked as skipped.

9.5.1. Image Files

In order for LT to load and display images, they must be stored in the `images` sub-

directory of the project directory, and they must be in **Utiff(5)** format. See the man page for **ras2ut(1)** for information about how to convert raster images to **Utiff(5)** format.

Probably the most important thing to know about image files is that LT never changes them.

9.5.2. Feature Data Files

Each type of feature data is stored in a subdirectory of the project directory. All feature data is in ASCII format, and is associated with a particular image by file naming conventions. LT reads all of the feature data associated with a pair of images after it reads the image data. When the user chooses **Save** from the stereo window **File** menu, LT writes the data back to the files, destroying the old information.

For grids points, there is an additional file containing the 0-based row and column indices of grids points that were skipped.

9.5.3. The Resource File

The resource file must be named **Utchat**. The X Windows resource manager will search for **Utchat** in its directory search path at startup time. The best thing to do is install this file in `/usr/X11/app-defaults`, but you can also install a copy in each user's home directory, or put it in some known location and have each user's `XAPPLRESDIR` environment variable point to the directory that contains it.

9.5.4. The Grid Files

The grid files are stored in the `grids` subdirectory of the project directory. They are read at window creation time, and written when the user performs a **Save** operation in the same way that feature data files are.

9.6. File Naming Conventions

All image files and feature data files are named `img_<image id>`. The image id consists of a flightpath number and an image number separated by a period. Suppose that the user has asked LT to display images named `img_05.1` and `img_06.1`. After reading the image data from files:

```
$(SSCS_PROJ_DIR)/images/img_05.1 and  
$(SSCS_PROJ_DIR)/images/img_06.1,
```

LT will search the `breaks`, `drains`, `grids`, `margins`, `random` and `ties` subdirectories of the project directory for any files named `img_05.1` or `img_06.1`. If they are found, it will read and display the feature data from those files.



The `img_` prefix for file names is an X resource stored in the `Utchat` resource file.

The grids points skip file is kept in the grids subdirectory of the project directory and is named `s<image id>_<image id>`.

The grid definition file is kept in the grids subdirectory of the project directory and is named `g<image id>_<image id>`.

9.7. File Formats

Image files are stored in `Utiff(5)` format. The resource file is in standard X resource format.

There are two classes of feature data, each having its own file format. All data for feature types which are single points (margin points, grids points, random points, and tie points) is stored in ASCII files. Each line contains the image space coordinates of one point, plus its unique ID:

```
<x><whitespace><y><whitespace><id>
```

where `<x>` and `<y>` are floating point numbers in `%f` format (see `printf (2)`), `<whitespace>` is a single tab character (though it could be any number of spaces), and `<id>` is an integer.

Data for feature types which are polylines (breaklines and drainlines) is stored in ASCII files as a series of line records. Each line record has this format:

```
<line header>
<point record>
<point record>
.
.
.
```

Where `<line header>` is a single `>` (right angle bracket) character followed by white space and an integer ID for the line. The point record format is the same as the one used for single point data.

The grids points skip file contains a list of skip records, each having this format:

```
<row><whitespace>< column>
```

where `<row>` and `<column>` are integers representing the 0-based row and column numbers of a skipped grid point.



The grid definition file contains six lines which can be in any order. Here's a sample grid definition file:

```
work region x: 2584
work region y: 3316
work region width: 1012
work region height: 5267
grid spacing x: 90
grid spacing y : 45
```

The work region x and y fields are the left-eye image coordinates of the upper-left corner of the grid. The work region width and height fields are the extents of the grid. The grid spacing x and y fields are the vertical and horizontal spacing between adjacent grid intersections.

10. Troubleshooting

This section contains a list of common problems and how to fix them.

Problem: The left- and right-eye images are reversed – moving the **Right H-Pan** dial moves the left-eye image, etc.

Solution: Change the **Depth Reversal** switch on the front of the stereo monitor. LT often comes up reversed or changes when you switch to the mono screen and back again. This is a hardware problem.

Problem: The screen contents look wrong – pieces of the images are misplaced, etc.

Solution: Bring up the window manager menu and refresh the screen (talk to your system administrator about how to set this up).

11. Shortcuts

LT has been designed so that all functionality is available from the menu bar, but there are several functions that have shortcuts for the experienced user.

11.1. Acquisition Popup Menu

Hold down the right mouse button to pop up the **Acquire** submenu of the stereo window **Edit** menu at the cursor position. For example, you can use this to change from Select mode to acquiring grid points without leaving the image window.

11.2. Arrow Keys

Pressing the up- and down-arrow keys while the dot cursor is in the image window adjusts the image x-parallax by one pixel per keystroke.

11.3. Accelerators

Several menu selections have *accelerators* attached to them, which are simple combinations of keystrokes that perform the same function as the menu selection.

Whenever there is an accelerator for a menu selection, the accelerator text is printed on the menu selection. For example, the accelerator for **Delete** (from the **Edit** menu) is ctrl-d, so the **Delete** button looks like this:

Delete ctrl-d

Here is a list of the accelerators currently defined:

- Delete – ctrl-d
- Undo – ctrl-u
- Save – ctrl-s
- Quit – ctrl-q

12. Customizing LT

For those familiar with X Windows, this is a list of resources that can be safely changed in the `Utchat` resource file to customize LT. The values listed here are the current defaults.

12.1. Dial Resources

```
lt*XmDial.foreground: MidnightBlue
lt*XmDial.indicatorColor: red
```

These control the appearance of the dials in the dial box. The `indicatorColor` resource controls the color of the dial “needle”.

```
lt*XmDial.numDialTicks: 20000
```

This controls the sensitivity of the dials. Higher numbers make the dials less sensitive (more movement of the dial produces less change in the dial value), and vice versa.

12.2. Feature Colors

```
lt.marginPointColor: cyan
lt.gridPointColor: red
lt.tiePointColor: green
lt.randomPointColor: yellow
lt.breakLineColor: magenta
lt.drainLineColor: blue
```

These resources control the color of the various types of feature marks. The values must exist in the X color database.

12.3. Feature Mark Sizes

```
lt.drainLinePointWidth: 6
lt.drainLinePointHeight: 4
lt.breakLinePointWidth: 6
lt.breakLinePointHeight: 4
```

These control the size of the Xs drawn by LT. You can set them to 0 to see smooth lines.

12.4. Project Directory Resources


```
lt.marginPointSubdirectory: margins  
lt.gridPointSubdirectory: grids  
lt.tiePointSubdirectory: ties  
lt.randomPointSubdirectory: random  
lt.drainLineSubdirectory: drains  
lt.breakLineSubdirectory: breaks
```

If for some reason you want to change the structure of the project directory, you can change these resources. They control where LT will look for feature data files. These subdirectory names are appended to the project directory name when assembling the feature data file names.

```
lt.imageFilePrefix: img_
```

This is the string that LT prepends to the image ID when assembling image file names.

```
lt.featureFilePrefix: img_
```

This is the string that LT prepends to the image ID when assembling feature data file names.

12.5. Accelerators

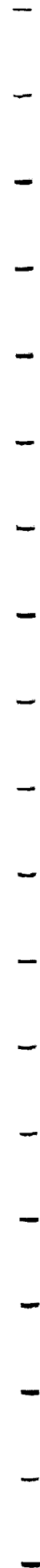
You can change the accelerators for menu selections by changing the

accelerator and
acceleratorText

resources for the appropriate menu buttons.

ATTACHMENT D

MODULES DESCRIPTION DOCUMENT



Vexcel Corporation

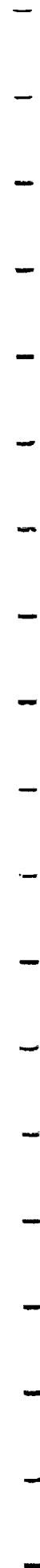
**EOS Workstation
modules description**

Author: Michel Millot
Last Revision: 10-17-91

1. INTRODUCTION.....	1
2. MODULES DESCRIPTIONS.....	1
2.1. DEM.....	1
2.1.1. TRANSFORM_CONTOURS.....	1
2.1.2. CONVERT_DEM.....	1
2.1.3. CONVERT_DEM2.....	2
2.1.4. EXTRACT_DEM.....	2
2.1.5. FILL_HOLES.....	3
2.1.6. READ_HEADER_DEM.....	4
2.1.7. MIX_DEM_DI.....	4
2.1.8. READ_DEM.....	5
2.1.9. ROTATE_DEM.....	5
2.1.10. FILL_HOLES_30.....	6
2.1.11. INTERP_DEM.....	6
2.1.12. READ_DEM_30.....	7
2.2. DLG.....	8
2.2.1. INTERP_HEIGHT_LINE.....	8
2.2.2. READ_LINE.....	8
2.2.3. FILTER_XYZ.....	9
2.2.4. RASTER_DLG.....	9
2.2.5. REM_PAIR_XYZ.....	10
2.3. NHAP.....	10
2.3.1. GENERATE_RESAMP_DATA.....	10
2.3.2. ROTATE_IMAGE.....	11
2.3.3. WRITE_TAPE.....	11
2.3.4. SIMCAM.....	12
2.4. TM.....	13
2.4.1. READ_TAPE.....	13
2.5. VISUALIZATION.....	13
2.5.1. CUT.....	13
2.5.2. DISPTXT.....	14
2.5.3. RUNME (imgslice).....	14
2.5.4. RUNME (newslice).....	14
2.5.5. CUT (tm_math).....	15
2.5.6. RUNME (variable).....	16
2.5.7. CUT (zoom3).....	16
2.6. SPOT.....	17
2.6.1. APPLY_POLY.....	17
2.6.2. ADDBD.....	18
2.6.3. APPLY_UTM_SPOT.....	18
2.6.4. CREATE_PHOTO.....	19
2.6.5. GENERATE_DATA.....	19
2.6.6. IMAGE_IMAGE.....	20
2.6.7. IMAGE_MAP.....	21
2.6.8. READ_GCP.....	21
2.6.9. READ_CHANNEL.....	22
2.6.10. REVERSE_POLY.....	22
2.6.11. READ_LEADER.....	23
2.6.12. REMOVE_BD.....	23
2.6.13. REVERSE_POLY_LT.....	24
2.6.14. WRITE_TAPE.....	24
2.6.15. GENERATE_RESAMP_DATA (Spot).....	24



2.6.16. GENERATE_RESAMP_DATA_DEM.....	25
2.6.17. GENERATE_RESAMP_DATA_DEMROT.....	26
2.6.18. ATTITUDE.....	26
2.6.19. ORBIT.....	27
2.6.20. PRESPT.....	27
2.6.21. SAT_SAT.....	27
2.6.22. SAT_UTM.....	27
2.6.23. SPT.....	28
2.6.24. UNDUL.....	28
2.6.25. UTM_SAT.....	28
2.7. MISC.....	29
2.7.1. ADD.....	29
2.7.2. CONV.....	29
2.7.3. CONV3D.....	29
2.7.4. CONV_SEQ.....	30
2.7.5. DEM_TRI.....	30
2.7.6. DEM_TRI_HIGH.....	31
2.7.7. MAX_TRI.....	32
2.7.8. MAX_TRI_HIGH.....	32
2.7.9. MEAN_SUB.....	33
2.7.10. RAWTOAIF.....	33
2.7.11. EXTVOL.....	33
2.7.12. PCAVOLDSK.....	34
2.7.13. REMOVE_BYTES.....	34
2.7.14. STRETCHING.....	35
2.7.15. STRIPAVS.....	35
2.7.16. USGS_TRI.....	35
3. PROCEDURES DESCRIPTION.....	36
3.1. PERSPECTIVE VIEW GENERATION.....	36
3.2. PAIR OF NHAP ORTHO IMAGE GENERATION.....	37
3.3. PAIR OF SPOT MULTISPECTRAL ORTHO IMAGES GENERATION.....	38
3.4. AIF IMAGE CUBE GENERATION.....	39
3.5. RASTERIZE A DLG SET.....	39
3.6. PROJECT DLG POINTS INTO SPOT IMAGES.....	39
3.7. PROJECT DLG POINTS INTO NHAP IMAGES.....	40
3.8. GENERATE ICE MOVEMENT.....	40



1. INTRODUCTION

This document describes the modules written during the prototyping phase of the EOS workstation project and the procedures used to generate the demonstrations. We choose to describe only the main executable modules and not the associated modules and we concentrate on the module inputs, outputs and purpose rather than on the implementation details.

2. MODULES DESCRIPTIONS

2.1. DEM

2.1.1. TRANSFORM_CONTOURS

MODULE: TRANSFORM_CONTOUR

LOCATION: VAX::_DUA2:[EOS.CONTOURS.TOOLS]

PURPOSE:

This program transforms an ASCII contour file, output of CPS3, to make it compatible with the program APPLY_UTM_SPOT

USES:

CPS3 contour line file

OUTPUTS:

ASCII file, containing:

- the number of contour lines
- for each contour line a unique id and the number of points
- for each point the X, Y, Z coordinate

2.1.2. CONVERT_DEM

MODULE: CONVERT_DEM

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Converts a DEM given in geographical coordinates into a DEM in UTM coordinates. The height at the DEM UTM nodes is computed by bilinear interpolation in the input DEM.

USES:

Input DEM defined as:

- first record containing:
 - . Lower left corner geographical coordinates
 - . Number of nodes
 - . Grid steps



- other records containing heights for a constant longitude

OUTPUTS:

Output DEM defined as:

- first record containing:
 - . Upper left corner UTM coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a constant Northing coordinate

2.1.3. CONVERT_DEM2

MODULE: CONVERT_DEM2

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Same as CONVERT_DEM, except that the ordering is the same for the input and output DEM. Consequently it runs faster

USES:

Input DEM defined as:

- first record containing:
 - . Lower left corner geographical coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a constant longitude

OUTPUTS:

Output DEM defined as:

- first record containing:
 - . Lower left corner UTM coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a constant Easting coordinate

2.1.4. EXTRACT_DEM

MODULE: EXTRACT_DEM

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Extract a window in a DEM stored in binary format and output



an ASCII file. It should be noted that there is no interpolation involved in this program, therefore the user provided window definition is fitted to the input DEM nodes location

USES:

Input DEM defined as:

- first record containing:
 - . Lower left or upper left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a column or a row
- Window definition:
 - . minimum X
 - . maximum X
 - . minimum Y
 - . maximum Y

OUTPUTS:

Output DEM window defined as:

- first record containing:
 - . 4 corners coordinates
 - . resolution in the 3 axis (grid step for the planimetric coordinates)
- other records containing the heights for constant Y coordinate

2.1.5. FILL_HOLES

MODULE: FILL_HOLES

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

In a given DEM, replace unknown values by interpolated values from another DEM. The interpolation used is bilinear interpolation

USES:

input DEM, with unknown values, defined as:

- first record containing:
 - . Lower left or upper left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a row

input DEM, to be used to fill the holes , defined as:

- first record containing:
 - . Lower left or upper left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a row



OUTPUTS:

input DEM with unknown values replaced by interpolated values

2.1.6. READ_HEADER_DEM

MODULE: READ_HEADER_DEM

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

read and display the header of a DEM stored in binary format

USES:

input DEM defined as:

- first record containing:
 - . Lower left or upper left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a row or a column

OUTPUTS:

displayed header:

- origin X coordinate
- origin Y coordinate
- X grid step
- Y grid step
- X number of nodes
- Y number of nodes

2.1.7. MIX_DEM_DI

MODULE: MIX_DEM_DI

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Merge two DEMs in binary format with the following properties (satisfied by USGS 1 degree DEM):

- same storage order
- origin at the upper left corner
- same definition along the X axis
- the last row of one is the first row of the other

USES:

2 input DEMs defined as:

- first record containing:
 - . upper left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a row

OUTPUTS:

- 1 merged DEM defined as:
 - first record containing:
 - . Upper left corner coordinates
 - . Number of nodes
 - . Grid steps
 - other records containing heights for a row

2.1.8. READ_DEM

MODULE: READ_DEM

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Read a USGS 1 degree DEM (on disk) and generate a DEM in binary format (see USGS "Digital Elevation Models Users Guide" for a format description)

USES:

USGS DEM

OUTPUTS:

- DEM, stored in binary format, defined as:
- first record containing:
 - . Lower left corner coordinates
 - . Number of nodes
 - . Grid steps
 - other records containing heights for a column

2.1.9. ROTATE_DEM

MODULE: ROTATE_DEM

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Read a DEM, with the origin at the lower left corner and stored by column, and generate a DEM, with the origin at the upper left corner and stored by line.

USES:

- DEM, stored in binary format, defined as:
- first record containing:
 - . Lower left corner coordinates
 - . Number of nodes
 - . Grid steps
 - other records containing heights for a column

OUTPUTS:

- DEM, stored in binary format, defined as:
- first record containing:
 - . Upper left corner coordinates
 - . Number of nodes



- . Grid steps
- other records containing heights for a row

2.1.10. FILL_HOLES_30

MODULE: FILL_HOLES_30

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

In a input DEM, replace unknown values by interpolated values using other DEMs. This program was written to compensate for the irregular definition of USGS 7.5 minute DEM (see USGS "Digital Elevation Models Users Guide")

USES:

Input DEM, stored in binary format, defined as:

- first record containing:
 - . Lower left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a column

Variable number of interpolation DEMs, stored in binary format, defined as:

- first record containing:
 - . Lower left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a column

OUTPUTS:

DEM, with the same definition as the input DEM stored in binary format, defined as:

- first record containing:
 - . Lower left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a column

2.1.11. INTERP_DEM

MODULE: INTERP_DEM

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Generate a new DEM from an input DEM. The new DEM heights are computed, by bilinear interpolation, from the input DEM heights. It also compute a resampling grid definition file

to be used by the SSCS resampling procedure.

USES:

- Input DEM, stored in binary format, defined as:
 - first record containing:
 - . Upper left corner coordinates
 - . Number of nodes
 - . Grid steps
 - other records containing heights for a row
- User inputs:
 - output DEM definition:
 - . origin X coordinate
 - . origin Y coordinate
 - . Grid step
 - . X number of nodes
 - . Y number of nodes
 - ortho image pixel size

OUTPUTS:

- DEM, stored in binary format, defined as:
 - first record containing:
 - . Upper left corner coordinates
 - . Number of nodes
 - . Grid steps
 - other records containing heights for a row
- resampling grid definition file defined as:
 - Upper left corner coordinates
 - X and Y number of nodes
 - pixel size
 - grid step

2.1.12. READ_DEM_30

MODULE: READ_DEM_30

LOCATION: VAX::_DUA2:[EOS.DEM.TOOLS]

PURPOSE:

Read a USGS 7.5 minute DEM and generate a DEM stored in binary format (see USGS "Digital Elevation Models Users Guide" for a format description). It should be noted that the USGS 7.5 minute DEM are not rectangular, but trapezoidal, therefore leading to unknown values in the rectangle containing the trapeze

USES:

USGS 7.5 minute DEM

OUTPUTS:

DEM, corresponding to the rectangle containing the input DEM and stored in binary format, defined as:

- first record containing:
 - . Lower left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a column

2.2. DLG

2.2.1. INTERP_HEIGHT_LINE

MODULE: INTERP_HEIGHT_LINE

LOCATION: VAX::_DUA2:[EOS.DLG.TOOLS]

PURPOSE:

For the processed DLG lines, compute the heights corresponding to the lines nodes planimetric coordinates. If one point is outside the DEM this module affects -999 as the unknown value

USES:

DEM, stored in binary format, defined as:

- first record containing:
 - . Upper left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a row

Processed DLG file containing:

- the number of lines
- for each line:
 - . the line number, line number of points
 - . for each point:
 - . X, Y coordinates

OUTPUTS:

3D DLG file containing:

- the number of lines
- for each line:
 - . the line number, line number of points
 - . for each point:
 - . X, Y, Z coordinates

2.2.2. READ_LINE

MODULE: READ_LINE

LOCATION: VAX::_DUA2:[EOS.DLG.TOOLS]

PURPOSE:

Read a USGS 1:100,000 DLG file and generate a processed DLG file (see USGS "Digital Line Graphs from 1:100,000-



Scale Maps Data Users Guide" for a format description)
USES:

USGS 1:100,000 DLG file OUTPUTS:

Processed DLG file containing:

- the number of lines
- for each line:
 - . the line number, line number of points
 - . for each point:
 - . X, Y coordinates

2.2.3. FILTER_XYZ

MODULE: FILTER_XYZ

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

In each line, in a set of lines, remove points with a distance (from the previous point) larger than a specified user specified threshold. This module may be useful to filter automatically some digitizing errors in the USGS DLG files. The User specified threshold should have the same unit as the points coordinates

USES:

- 3D DLG file containing:
 - the number of lines
 - for each line:
 - . the line number, line number of points
 - . for each point:
 - X, Y, Z coordinates
- User input of the distance threshold

OUTPUTS:

processed 3D DLG file containing:

- the number of lines
- for each line:
 - . the line number, line number of points
 - . for each point:
 - X, Y, Z coordinates

2.2.4. RASTER_DLG

MODULE: RASTER_DLG

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Transform a DLG file, in vector format, into a raster image. This module provides a flag allowing the user to add to an existing image the DLG lines. It should be noted that the image is a byte image and the lines are allocated the intensity 255

USES:

- 3D DLG file containing:

- the number of lines
- for each line:
 - . the line number, line number of points
 - . for each point:
 - X, Y, Z coordinates
- image definition in the points reference system:
 - . upper left corner planimetric coordinates
 - . number of pixels
 - . number of lines
 - . pixel size
- flag: 0 add to existing image, 1 new image

OUTPUTS:

- raster image

2.2.5. REM_PAIR_XYZ

MODULE: REM_PAIR_XYZ

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Remove lines with only 2 points

USES:

- 3D DLG file containing:
 - the number of lines
 - for each line:
 - . the line number, line number of points
 - . for each point:
 - X, Y, Z coordinates

OUTPUTS:

- 3D DLG file containing:
 - the number of lines
 - for each line:
 - . the line number, line number of points
 - . for each point:
 - X, Y, Z coordinates

2.3. NHAP

2.3.1. GENERATE_RESAMP_DATA

MODULE: GENERATE_RESAMP_DATA

LOCATION: VAX::_DUA2:[EOS.NHAP.MISC]

PURPOSE:

From a DEM file and the corresponding NHAP image coordinates file, generate the line and pixel grid files. This



module assumes a one to one correspondence between a DEM posting and a pixel location. It formats the data to allow the use of the SSCS resampling procedure

USES:

ASCII DEM file containing:
. for each node the X, Y, Z coordinates

NHAP image coordinates file containing:
. for each DEM node, the corresponding pixel and line number

OUTPUTS:

line grid file containing:
. for each DEM posting, the Y, X coordinates and the corresponding line number

pixel grid file containing:
. for each DEM posting, the Y, X coordinates and the corresponding pixel number

2.3.2. ROTATE_IMAGE

MODULE: ROTATE_IMAGE

LOCATION: VAX::_DUA2:[EOS.NHAP.ROTATE]

PURPOSE:

Rotate an image in a BD format. More precisely the user chooses among the following options:

- Rotate image upon X axis
- Rotate image upon Y axis
- Rotate image across the diagonal
- Rotate image upon X axis and orthogonal to the origin
- Rotate image upon Y axis and orthogonal to the origin
- Rotate image across diagonal and orthogonal to origin
- Rotate image orthogonal to origin

this module may be used when aerial photography are digitized inconsistently with the flight direction, therefore impeding stereo perception and acquisition

USES:

input image in BD format (see Vexcel documentation) User rotation choice

OUTPUTS:

output image in BD format

2.3.3. WRITE_TAPE

MODULE: WRITE_TAPE

LOCATION: VAX::_DUA2:[EOS.NHAP.WRITE_TAPE]

PURPOSE:

Write one or several BD format image files on a tape. This module do not handle multiple tapes

USES:

one or several image files in BD format (see VEXCEL documentation)

User inputs:

- . rewind the tape
- . skip files on tape
- . end of file

OUTPUTS:

input files on tape in simple raster format (no header, no record prefix or suffix, no trailer)

2.3.4. SIMCAM

MODULE: SIMCAM

LOCATION: jupiter:/u0/home/michel/dave/src

PURPOSE:

Camera simulation program. It reads camera relevant parameters (exterior orientation and interior transform) and apply them to 3D coordinates, to output image coordinates. The camera parameters are computed using PHOTOG and do not require the control points acquisition on line inside PHOTOG

USES:

- Camera parameters file:
 - . exterior orientation:
 - Translation vector X coordinate
 - Translation vector Y coordinate
 - Translation vector Z coordinate
 - Omega angle
 - Phi angle
 - Kappa angle
 - . Calibration:
 - focal length
 - principal point coordinates
 - . interior transform
 - translation vector X coordinate
 - translation vector Y coordinate
 - 2D rotation matrix
- Points ground coordinates file:
 - . X coordinate
 - . Y coordinate
 - . Z coordinate

OUTPUTS:

- points image coordinates file:
 - . pixel number
 - . line number

2.4. TM

2.4.1. READ_TAPE

MODULE: READ_TAPE

LOCATION: VAX::_DUA2:[EOS.TM.TOOLS]

PURPOSE:

Read, from tape, a Thematic Mapper image and output a set of raster files; one for each spectral band. This module also allows the user to define a window. It should be noted that this module do not manage multiple tapes and has no knowledge of tape structure

USES:

- User input of:
 - . number of spectral bands to read
 - . the image dimension:
 - number of lines
 - number of columns
 - . the window definition:
 - origin coordinates
 - window dimension:
 - . number of lines
 - . number of pixels
 - . tape blocking factor

OUTPUTS:

spectral bands raster files

2.5. VISUALIZATION

2.5.1. CUT

MODULE: CUT (cut)

LOCATION: jupiter:/u0/home/michel/dore/cut

PURPOSE:

Display 7 images in a CUT and PASTE mode. The first image is used as the background image, the remaining 6 images are displayed inside a window which position is controlled by the mouse position. The choice between the PASTE images is given to the user through the mouse buttons and shift key. It should be noted that this module do not allow images larger than the screen size, works only for 8 bits images and expects images of the same dimensions and is Stardent specific (needs low level Stardent functions)

USES:

- 7 images in aif format; header, then image (see Stardent



documentation)

- User input of:

- . window width

- . window height

- Panel ASCII file, containing a description of the data and the module. At execution time, It is displayed as an introductory message and is erased by the user. The first line is considered a title and consequently is displayed using a different font

- Label ASCII file, Containing the link between mouse events and images. During the execution, these labels are displayed on the right side of the screen in the area not used for the images

OUTPUTS:

None

2.5.2. DISPTXT

MODULE: DISPTXT

LOCATION: jupiter:/u0/home/michel/dore/disptxt

PURPOSE:

Display on the screen a text stored in an ASCII file, using X Windows. The first line is considered a title and consequently is displayed using a different font.

USES:

text ASCII file

OUTPUTS:

None

2.5.3. RUNME (imgslice)

MODULE: RUNME

LOCATION: jupiter:/farm/kelly/Stardent_demos/imgslice

PURPOSE:

Modified version of the Stardent 3D image cubes visualization tool. It improves the existing module by providing labels with each image cube layer. These labels are always displayed and follow the image scale. It should be noted that this module expects an image cube with a 512x512 spatial size

USES:

- image cube, composed of 8 bits layers in aif format. It should be declared in the corresponding data.button file at the file button definition

- ASCII label file (one layer description record per layer). It must have the same number of records as the image cube number of layers and must have the same name as the image cube with the extension being '.lab'

OUTPUTS:

None

2.5.4. RUNME (newslice)

MODULE: RUNME

LOCATION: jupiter:/farm/kelly/Stardent_demos/newslice

PURPOSE:

Modified version of the Stardent 3D image cubes visualization tool. It improves the existing module by providing labels with each image cube layer and by allowing the user to input one image cube at two different resolutions. The labels are always displayed and follow the image scale. It should be noted that this module expects the low resolution version image cube to be 256x256 and the high resolution version to be 512x512

USES:

- low resolution image cube, composed of 8 bits layers in aif format. It should be declared in the corresponding data.button file at the file button definition
- high resolution image cube, composed of the same 8 bits layers as the low resolution image cube, but with the resolution doubled. It must have the same number of layers as the low resolution image cube and must have the same name as the low resolution image cube with the extension being '.ai2'
- ASCII label file (one layer description record per layer). It must have the same number of records as the image cube number of layers and must have the same name as the low resolution image cube with the extension being '.lab'

OUTPUTS:

None

2.5.5. CUT (tm_math)

MODULE: CUT

LOCATION: /u0/home/michel/dore/tm_math

PURPOSE:

Similar to CUT in /u0/home/michel/dore/cut, but Display 4 images in a CUT and PASTE mode. The first image is used as the background image, the remaining 3 images are displayed inside a window which position is controlled by the mouse position. The choice between the PASTE images is given to the user through the mouse buttons and shift key. It should be noted that this module do not allow images larger than the screen size, works only for 8 bits images and expects images of the same dimensions and is Stardent specific (needs low level Stardent functions)

USES:

- 4 images in aif format; header, then image (see Stardent documentation)
- User input of:
 - . window width
 - . window height
- ASCII panel file, containing a description of the data and the module. At execution time, It is displayed as an introductory

message and is erased by the user. The first line is considered a title and consequently is displayed using a different font
- ASCII label file, Containing the link between mouse events and images. During the execution, these labels are displayed on the right side of the screen in the area not used for the images

OUTPUTS:

None

2.5.6. RUNME (variable)

MODULE: RUNME

LOCATION: jupiter:/u0/home/michel/dore/variable

PURPOSE:

Modified version of the Stardent perspective view visualization tool. It improves the existing module by providing an additional dial widget (identified as "gain" dial) to adjust the Z component relative to the X, Y components. It should be noted that the 2 mesh files name are defined inside the geom_spec module.

USES:

- low resolution mesh file. It contains the triangle mesh representing the surface to be displayed. More precisely:
 - . number of vertices
 - . for each vertex:
 - X coordinate
 - Y coordinate
 - Z coordinate
 - . number of triangles
 - . for each triangle:
 - 3 vertices indices in counterclockwise order
- high resolution mesh file. It contains the triangle mesh representing the surface to be displayed with a color (defined as RGB) associated with each vertex. More precisely:
 - . number of vertices
 - . for each vertex:
 - X coordinate
 - Y coordinate
 - Z coordinate
 - Red value
 - Green value
 - Blue value
 - . number of triangles
 - . for each triangle:
 - 3 vertices indices in counterclockwise order

OUTPUTS:

None

2.5.7. CUT (zoom3)

MODULE: CUT

LOCATION: jupiter:/u0/home/dore/zoom3

PURPOSE:

Display 2 images, with different nominal resolution, in a CUT and PASTE mode. The first image is used as the background image, the second image, displayed inside a moving window, is displayed at 3 different resolutions. It is assumed that the second image first scale is the same as the first image, the second image second scale is 2 times larger than the first image and the second image third scale is 4 times larger than the first image. In addition to this built-in zooming capabilities, this module provides continuous zooming of:

- the first image, from 1 to 4 times the original scale
- the second image, from 1 to 4 times the first image scale
- a weighted sum of the first image and second image, from 1 to 4 times the first image scale, with the weight proportional to the zooming factor

The choice between the images is given to the user through the mouse buttons and shift key. It should be noted that this module do not allow the first image to be larger than the screen size, works only for 8 bits images, is Stardent specific (needs low level Stardent functions) and has the explanation panel and link between the mouse events and images displayed built-in

USES:

- first image file in aif format; header, then image (see Stardent documentation)
- second image first scale file in aif format, this image should have the same dimensions as the first image
- second image second scale file in aif format
- second image third scale file in aif format
- User input of:
 - . window width
 - . window height

OUTPUTS:

None

2.6. SPOT

2.6.1. APPLY_POLY

MODULE: APPLY_POLY

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Transform 2D coordinates into 2D coordinates, using previously estimated polynomial coefficients. For example for a first degree polynomial, the coefficients are stored as:

- . first coordinate coefficient
- . second coordinate coefficient



. constant

USES:

- Polynomial degree
- first coordinate polynomial coefficients file
- second coordinate polynomial coefficients file
- input points file defined as:
 - point 2D coordinates

OUTPUTS:

- output points file defined as:
 - point 2D coordinates

2.6.2. ADDBD

MODULE: ADDBD

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Transform a raster image file into a BD image file. The input image intensities may be 1 byte or 2 bytes long.

USES:

- User input of:
 - . 1 or 2 bytes word
 - . input image number of line and columns
- input image in raster format

OUTPUTS:

output image in BD format

2.6.3. APPLY_UTM_SPOT

MODULE: APPLY_UTM_SPOT

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Compute the SPOT images coordinates of a variable number of 3D DLG files. The transformation, from (E,N,h) to (l1,p1) and (l2,p2) was estimated using SPOTCHECK. it should be noted that this module is independent of the cartographic projection used, as long as the 3D DLG points are given in the same projection as the one used in the transformation estimation. This module was written for 2 SPOT images

USES:

- file containing the first degree polynomial coefficients mapping the transformation from SPOTCHECK reference system to the original SPOT images reference system
- file containing the polynomial coefficients mapping the transformation from the SPOTCHECK images reference system to the cartographic reference system
- file containing the polynomial coefficients mapping the

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

transformation from the cartographic reference system to the SPOTCHECK images reference system

- variable number of 3D DLG files defined as:

- the number of lines

- for each line:

- . the line number, line number of points

- . for each point:

- . X, Y, Z coordinates

- User input of the SPOT images number of lines and number of pixels (should be identical for the 2 images)

OUTPUTS:

- file containing the image 1 coordinates of all the DLG 3D lines with more than 3 points inside the image

- file containing the image 2 coordinates of all the DLG 3D lines with more than 3 points inside the image these files are defined as:

- . line separator is '>', line number

- . each point record is composed of pixel number, line number, point number (relative to the line)

2.6.4. CREATE_PHOTO

MODULE: CREATE_PHOTO

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Generate the ground control points and homologous points images coordinates files in SPOTCHECK format

USES:

- left image coordinates file, organized as:

- . point number, pixel number, line number

- right image coordinates file, organized as:

- . point number, pixel number, line number

OUTPUTS:

SPOTCHECK control points image coordinates file, organized as:

- first 4 records containing the image 4 corners coordinates (origin lower left corner and after clockwise)

- other records containing:

- . point number

- . right image pixel number

- . right image line number

- . left image pixel number

- . left image line number

2.6.5. GENERATE_DATA

MODULE: GENERATE_DATA

—

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Generate the files needed to resample an image using the SSCS procedure. The mapping between the output reference system and the original image is represented by a polynomial of arbitrary degree

USES:

- User inputs:
 - . output image number of pixels
 - . output image number of lines
 - . resampling grid step
 - . polynomial degree
- line polynomial coefficients file. For example for a first degree polynomial, the coefficients are stored as:
 - . first coordinate coefficient
 - . second coordinate coefficient
 - . constant
- pixel polynomial coefficients file

OUTPUTS:

- resampling grid definition file containing:
 - . origin coordinates (assumed to be (1,1))
 - . X, Y number of nodes
 - . pixel size (assumed to be 1)
 - . grid step
- line grid file containing:
 - . output image line number, output image pixel number, input image line number
- pixel grid file containing:
 - . output image line number, output image pixel number, input image pixel number

2.6.6. IMAGE_IMAGE

MODULE: IMAGE_IMAGE

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Estimate the mapping between 2 images. The mapping is represented by a polynomial of arbitrary degree and the polynomial estimation is done by least mean square. The estimated polynomial transform reference image coordinates into secondary image coordinates

USES:

- homologous points secondary image coordinates file containing:
 - . point number
 - . pixel number
 - . line number
- homologous points reference image coordinates file containing:
 - . point number

- . pixel number
- . line number
- User input of the polynomial degree

OUTPUTS:

- ASCII result file containing:
 - . For each homologous point:
 - . point number
 - . secondary image measured line number
 - . secondary image computed line number
 - . secondary image measured pixel number
 - . secondary image computed pixel number
 - . line and pixel polynomial coefficients
- line polynomial coefficients file
- pixel polynomial coefficients file

2.6.7. IMAGE_MAP

MODULE: IMAGE_MAP

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Estimate the mapping between one image and the ground. The mapping is represented by a polynomial of arbitrary degree and the polynomial estimation is done by least mean square. The estimated polynomial transform reference image coordinates into secondary image coordinates. It should be noted that the point's heights are not used for the mapping estimation.

USES:

- Ground control points image coordinates file containing:
 - . point number
 - . pixel number
 - . line number
- Ground control points ground coordinates file containing:
 - . point number
 - . X ground coordinate
 - . Y ground coordinate
 - . Z ground coordinate

OUTPUTS:

- ASCII result file containing:
 - . For each ground control point:
 - . point number
 - . image measured line number
 - . image computed line number
 - . image measured pixel number
 - . image computed pixel number
 - . line and pixel polynomial coefficients
- line polynomial coefficients file
- pixel polynomial coefficients file

2.6.8. READ_GCP

MODULE: READ_GCP

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Format ground coordinates. output of KORK (digitizing program) into ground coordinates compatible with the SPOTCHECK package

USES:

ground coordinates file in KORK format:

- . comments beginning with the '!' character
- . invalid points with the point number equal to 0
- . valid points record composed of:
 - point number
 - ground X coordinate
 - ground Y coordinate
 - ground Z coordinate

OUTPUTS:

ground coordinates file in SPOTCHECK format:

- . point number
- . ground X coordinate
- . ground Y coordinate
- . ground Z coordinate

2.6.9. READ_CHANNEL

MODULE: READ_CHANNEL

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Generate 3 files corresponding to each band of a SPOT multispectral image. It should be noted that the spectral bands are written on the tape, with the BIL (Band Interleaved per Line) format. Also the module assumes that the image is geometrically raw, therefore is with a nominal dimension of 3000x3000.

USES:

Spot Image Corporation original imagery file

OUTPUTS:

- 3 image files in raster format:
- . 1 record per image line

2.6.10. REVERSE_POLY

MODULE: REVERSE_POLY

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Apply the inverse of a first degree polynomial, given the direct polynomial coefficients. The polynomial is assumed to be of degree 1 to be consistent with the purpose of computing a



unique inverse, when the domain is not considered. The module current implementation assumes image coordinates but is by essence a general 2D module

USES:

- line polynomial coefficients file
- pixel polynomial coefficients file
- image coordinates file, containing for each pixel:
 - . point number
 - . pixel number
 - . line number

OUTPUTS:

- modified image coordinates file, containing for each pixel:
- . point number
 - . pixel number
 - . line number

2.6.11. READ_LEADER

MODULE: READ_LEADER

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

In the leader file (Spot Image Corporation format), read the useful image ancillary data. The usefulness is defined as the data required by SPOTCHECK.

USES:

Leader file in Spot Image Corporation format

OUTPUTS:

useful data:

- geographical coordinates of the scene center
- scene center time
- revolution number in the cycle
- satellite position
- satellite attitude

2.6.12. REMOVE_BD

MODULE: REMOVEBD

LOCATION: _DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

In an image in BD format, remove the header and modify the record size to the image line size. It allows the user to output an image window.

USES:

BD image file

OUTPUTS:

- User defined window:
 - window origin in the original image
 - window dimensions

- Raster file containing the window

2.6.13. REVERSE_POLY_LT

MODULE: REVERSE_POLY_LT

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Apply the inverse of a first degree polynomial, given the direct polynomial coefficients. The polynomial is assumed to be of degree 1 to be consistent with the purpose of computing a unique inverse, when the domain is not considered. The module current implementation assumes image coordinates but is by essence a general 2D module

USES:

line polynomial coefficients file pixel polynomial coefficients file LT image coordinates file, containing for each point:

- . pixel number
- . line number
- . point number

OUTPUTS:

modified image coordinates file, containing for each pixel:

- . pixel number
- . line number
- . point number

2.6.14. WRITE_TAPE

MODULE: WRITE_TAPE

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Write on a tape, one or several image files in BD format

USES:

one or several image file, in BD format

OUTPUTS:

updated tape with images in raster format

2.6.15. GENERATE_RESAMP_DATA (Spot)

MODULE: GENERATE_RESAMP_DATA

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Output the files needed to generate a SPOT geocoded image. The transformation, from (E,N,h) to (l1,p1) and (l2,p2) was estimated using SPOTCHECK. it should be noted that this module is independent of the cartographic projection used, as long as the resampling grid nodes coordinates are given in the same projection as the one used in the transformation estimation. This module was written for 2 SPOT images.

USES:

- file containing the first degree polynomial coefficients mapping the transformation from SPOTCHECK reference system to the original SPOT images reference system
- file containing the polynomial coefficients mapping the transformation from the SPOTCHECK images reference system to the cartographic reference system
- file containing the polynomial coefficients mapping the transformation from the cartographic reference system to the SPOTCHECK images reference system
- User input of:
 - minimum terrain height
 - maximum terrain height
 - reference height (height used in the ground to image mapping)
 - resampled image pixel size
 - resampling grid step

OUTPUTS:

- resampling grid definition file, containing:
 - . grid origin (1,1) planimetric coordinates
 - . number of nodes along the X and Y axis
 - . pixel size
 - . grid step
- for each image:
 - . line grid file containing:
 - . for each grid node, the Y, X coordinates and the corresponding line number
 - . pixel grid file containing:
 - . for each grid node, the Y, X coordinates and the corresponding pixel number

2.6.16. GENERATE_RESAMP_DATA_DEM

MODULE: GENERATE_RESAMP_DATA_DEM

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

Output the files needed to generate a SPOT ortho image. The transformation, from (E,N,h) to (l1,p1) and (l2,p2) was estimated using SPOTCHECK. it should be noted that this module is independent of the cartographic projection used, as long as the DEM posting coordinates are given in the same projection as the one used in the transformation estimation. This module was written for 2 SPOT images. The elevation at each resampling grid node is provided by bilinear interpolation in a DEM

USES:

- file containing the first degree polynomial coefficients mapping the transformation from SPOTCHECK reference system to the original SPOT images reference system
- file containing the polynomial coefficients mapping the transformation from the SPOTCHECK images reference system to the cartographic reference system



- file containing the polynomial coefficients mapping the transformation from the cartographic reference system to the SPOTCHECK images reference system
- User input of:
 - minimum terrain height
 - maximum terrain height
 - resampled image pixel size
 - resampling grid step
- Variable number of DEMs, with the following format:
 - first record containing:
 - . Lower left corner coordinates
 - . Number of nodes
 - . Grid steps
 - other records containing heights for a DEM column

OUTPUTS:

- resampling grid definition file, containing:
 - . grid origin (1,1) planimetric coordinates
 - . number of nodes along the X and Y axis
 - . pixel size
 - . grid step
- for each image:
 - . line grid file containing:
 - . for each grid node, the Y, X coordinates and the corresponding line number
 - . pixel grid file containing:
 - . for each grid node, the Y, X coordinates and the corresponding pixel number

2.6.17. GENERATE_RESAMP_DATA_DEMROT

MODULE: GENERATE_RESAMP_DATA_DEMROT

LOCATION: VAX::_DUA2:[EOS.SPOT.TOOLS]

PURPOSE:

See GENERATE_RESAMP_DATA_DEM. The only difference is the DEM storage which is identical to the resampling grid, therefore giving a significant performance improvement

USES:

Identical to GENERATE_RESAMP_DATA_DEM, except for the DEM storage:

- first record containing:
 - . Upper left corner coordinates
 - . Number of nodes
 - . Grid steps
- other records containing heights for a DEM line

OUTPUTS:

See GENERATE_RESAMP_DATA_DEM

2.6.18. ATTITUDE

MODULE: ATTITUDE

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Process the SPOT attitude data (See SPOTCHECK documentation)

USES:

See SPOTCHECK documentation

OUTPUTS:

See SPOTCHECK documentation

2.6.19. ORBIT

MODULE: ORBIT

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Process the SPOT ephemeris data (See SPOTCHECK documentation)

USES:

See SPOTCHECK documentation

OUTPUTS:

See SPOTCHECK documentation

2.6.20. PRESPT

MODULE: PRESPT

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Preprocess the ground control points (See SPOTCHECK documentation)

USES:

See SPOTCHECK documentation

OUTPUTS:

See SPOTCHECK documentation

2.6.21. SAT_SAT

MODULE: SAT_SAT

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Estimate a polynomial mapping the transformation between the 2 SPOT images (See SPOTCHECK documentation)

USES:

See SPOTCHECK documentation

OUTPUTS:

See SPOTCHECK documentation

2.6.22. SAT_UTM

MODULE: SAT_UTM

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Estimate a polynomial mapping the transformation from each of the SPOT stereo pair images to the cartographical reference (See SPOTCHECK documentation)

USES:

See SPOTCHECK documentation

OUTPUTS:

See SPOTCHECK documentation

2.6.23. SPT

MODULE: SPT

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Using ground control points, estimate corrections to the ancillary data of a SPOT stereo pair (See SPOTCHECK documentation)

USES:

See SPOTCHECK documentation

OUTPUTS:

See SPOTCHECK documentation

2.6.24. UNDUL

MODULE: UNDUL

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Using a geoid numerical approximation, Compute the corresponding height corrections between "geographical" and "physical" heights (See SPOTCHECK documentation)

USES:

See SPOTCHECK documentation

OUTPUTS:

See SPOTCHECK documentation

2.6.25. UTM_SAT

MODULE: UTM_SAT

LOCATION: jupiter:/u0/home/michel/spotcheck

PURPOSE:

Estimate a polynomial mapping the transformation from the cartographical reference to each of the SPOT stereo pair images (See SPOTCHECK documentation)

USES:

OUTPUTS: See SPOTCHECK documentation
See SPOTCHECK documentation

2.7. MISC

2.7.1. ADD

MODULE: ADD

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Add two raster images, in byte format, and generate the result image in byte format. If the sum exceeds the byte representation, the value is truncated to the nearest valid byte value. the two images must have the same dimension

USES:

- image 1 file
- image 2 file
- user input of:
 - . common number of pixels
 - . common number of lines

OUTPUTS:

output image file

2.7.2. CONV

MODULE: CONV

LOCATION: /u0/home/michel/utilities/src

PURPOSE:

Combine 3 images to generate an output image in BIP (Band Interleaved per Pixel) format. The 3 images are assumed to have the same dimensions and to follow the following naming convention; common name plus '.red', '.grn' or '.blu' extension. The output file is named after the input files name with the '.rgb' extension. It should be noted that the output image has 4 bytes per pixel, the alpha byte being set to 0.

USES:

- "red" image file
- "green" image file
- "Blue" image file
- User input of the images number of lines and number of pixels.

OUTPUTS:

- "rgb" image file

2.7.3. CONV3D

MODULE: CONV3D

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Combine 4 image cubes to generate an output image cube, organized in BIP (Band Interleaved per Pixel) format, for each spatial layer. The 3 image cubes are assumed to have the same dimensions and to follow the following naming convention; common name plus '.alp', '.red', '.grn' or '.blu' extension. The output file is named after the input files name with the '.rgb' extension

USES:

- "alpha" image cube
- "red" image cube
- "green" image cube
- "blue" image cube
- User input of the image cubes number of layers, number of pixels, number of lines

OUTPUTS:

"argb" image cube

2.7.4. CONV_SEQ

MODULE: CONV_SEQ

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Combine 3 images to generate an output image in BSQ (Band SeQuential) format. The 3 images are assumed to have the same dimensions and to follow the following naming convention; common name plus '.red', '.grn' or '.blu' extension. The output file is named after the input files name with the '.vol' extension. The user may also define a window for the output image.

USES:

- "red" image file
- "green" image file
- "blue" image file
- User input of:
 - . images number of pixels
 - . images number of lines
 - . window upper left coordinates
 - . window number of pixels
 - . window number of lines

OUTPUTS:

- "volume" image file

2.7.5. DEM_TRI

MODULE: DEM_TRI

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

From an AVS 2D scalar, byte or integer uniform field, Generate the corresponding triangle mesh. It should be noted that this module contains no optimization and therefore each pixel of the user specified window is a vertex in the output triangle mesh

USES:

- AVS 2D scalar, byte or integer, uniform field (See Stardent AVS documentation)
- User input of:
 - . window upper left corner coordinates
 - . window dimensions
 - . sampling factors, pixel and line, to apply inside the window

OUTPUTS:

triangle mesh with the following format:

- . number of vertices
- . for each vertex:
 - X coordinate
 - Y coordinate
 - Z coordinate
- . number of triangles
- . for each triangle:
 - 3 vertices indices in counterclockwise order

2.7.6. DEM_TRI_HIGH

MODULE: DEM_TRI_HIGH

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

From an AVS 2D scalar, byte or integer uniform field, containing the vertices elevation, and an AVS 2D vector 4 bytes uniform field, containing the vertices color, Generate the corresponding triangle mesh. It should be noted that this module contains no optimization and therefore each pixel of the user specified window is a vertex in the output triangle mesh and it requires the 2 AVS fields to have the same dimensions.

USES:

- AVS 2D scalar, byte or integer, uniform field (see Stardent AVS documentation)
- AVS 2D vector 4 bytes uniform field (see Stardent AVS documentation)
- User input of:
 - . window upper left corner coordinates
 - . window dimensions
 - . sampling factors, pixel and line, to apply inside the window

OUTPUTS:

triangle mesh, with the following format:

- . number of vertices
- . for each vertex:
 - X coordinate
 - Y coordinate

- Z coordinate
- Red value
- Green value
- Blue value
- . number of triangles
- . for each triangle:
 - 3 vertices indices in counterclockwise order

2.7.7. MAX_TRI

MODULE: MAX_TRI

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE: Compute the range of each vertex coordinate of a triangle mesh.

USES:

triangle mesh with the following format:

- . number of vertices
- . for each vertex:
 - X coordinate
 - Y coordinate
 - Z coordinate
- . number of triangles
- . for each triangle:
 - 3 vertices indices in counterclockwise order

OUTPUTS:

minimum and maximum of each vertex coordinate

2.7.8. MAX_TRI_HIGH

MODULE: MAX_TRI_HIGH

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE: Compute the range of each vertex coordinate of a triangle mesh.

USES:

triangle mesh, with the following format:

- . number of vertices
- . for each vertex:
 - X coordinate
 - Y coordinate
 - Z coordinate
 - Red value
 - Green value
 - Blue value
- . number of triangles
- . for each triangle:
 - 3 vertices indices in counterclockwise order

OUTPUTS:

minimum and maximum of each vertex coordinate

2.7.9. MEAN_SUB

MODULE: MEAN_SUB

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Reduce a raster image stored in BIP (Band Interleaved per Pixel) format. This module computes each output as the mean of the window centered at the output pixel location. The window size indicates also the reduction factor

USES:

- image file
- User input of:
 - . image number of pixels
 - . image number of lines
 - . reduction factor
 - . number of bytes per pixel

OUTPUTS:

- reduced image file

2.7.10. RAWTOAIF

MODULE: RAWTOAIF

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Transform a raster image into an image in aif format. The input image may be a 1 byte per pixel image or a 3 bytes per pixel (RGB) image.

USES:

- raster image
- User input of:
 - . image number of pixels
 - . image number of lines
 - . color/BW flag, 0: 1 byte per pixel, 1: 3 bytes per pixel

OUTPUTS:

- aif image

2.7.11. EXTVOL

MODULE: EXTVOL

LOCATION: /u0/home/kelly/eos/src

PURPOSE:

Extracts an image cube from an existing image cube. Additionally, replication or subsampling in any direction is supported. It should be noted that the image cube is expected to be in BSQ (Band SeQuential) format

USES:

- image cube

- User input of:
 - . number of layers
 - . number of lines
 - . number of pixels
 - . image cube window definition:
 - first layer
 - last layer
 - first line
 - last line
 - first pixel
 - last pixel
 - . sampling factor (plus indicates replication, minus subsampling):
 - layer
 - line
 - pixel
 - . image cube format; 0 byte, 1 I*2 . Header size, in bytes, to be stripped

OUTPUTS:

- image cube window (without header)

2.7.12. PCAVOLDSK

MODULE: PCAVOLDSK

LOCATION: /u0/home/kelly/eos/src

PURPOSE:

Perform a Principle Components Analysis (PCA) of an image cube in BSQ (Band SeQuential) format, without header. It should be noted that the module expects a byte image cube and, optionally, allows the user to specify the memory size to be allocated for the processing

USES:

- image cube
- User input of:
 - . number of layers
 - . number of lines
 - . number of pixels
 - . number of layers to be generated
 - . optionally, the memory size specified as a number of lines (the effective memory allocated is given by: number of lines*image cube number of layers*image cube number of pixels)

OUTPUTS:

- PCA image cube

2.7.13. REMOVE_BYTES

MODULE: REMOVE_BYTES

LOCATION: jupiter:/u0/home/michel/utilities

PURPOSE:

USES: Remove a user specified number of bytes from a binary file

- binary file

OUTPUTS:

- User input of the number of bytes to skip
- new binary file

2.7.14. STRETCHING

MODULE: STRETCHING

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Apply a linear ramp to the intensities of a byte raster image

USES:

- raster image
- User input of:
 - . image number of pixels
 - . image number of lines
 - . Ramp bottom (all intensities smaller or equal will be assigned 0 as output intensity)
 - . ramp top (all intensities greater or equal will be assigned 255 as output intensity)

OUTPUTS:

- stretched raster image

2.7.15. STRIPAVS

MODULE: STRIPAVS

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Remove the header of a 3D AVS field or AVS volume (See Stardent AVS documentation)

USES:

- AVS file
- User input of:
 - . number of layers
 - . number of lines
 - . number of pixels
 - . header type: 0 represents 3 bytes for an AVS volume, 1 represents a variable number of bytes for an AVS field

OUTPUTS:

- AVS file data

2.7.16. USGS_TRI

MODULE: USGS_TRI

LOCATION: jupiter:/u0/home/michel/utilities/src

PURPOSE:

Transform a 1 degree USGS DEM into a triangle mesh.
Additionally, this module subsample the input DEM

USES:

- 1 degree USGS DEM (see USGS "Digital Elevation Models Data Users Guide")
- User input of:
 - . sampling factor in columns and rows

OUTPUTS:

- triangle mesh, with the following format:
 - . number of vertices
 - . for each vertex:
 - X coordinate
 - Y coordinate
 - Z coordinate
 - Red value
 - Green value
 - Blue value
 - . number of triangles
 - . for each triangle:
 - 3 vertices indices in counterclockwise order

3. PROCEDURES DESCRIPTION

3.1. PERSPECTIVE VIEW GENERATION

INPUT DATA:

- ortho image. raster image BW or color, if color 3 distinct files for each color
- DEM. may be ASCII or binary but without header information.

REQUIREMENTS:

- the image and the DEM should be registered
- the distance between 2 DEM nodes should be a multiple of the image pixel size, for better quality a 1:1 ratio is recommended
- a maximum number of 150,000 triangles is recommended, for real time rendering
- the DEM should be stored and used in integer format to avoid the quantification generated with the [0-255] range
- to avoid confusion, the image and DEM should have the same origin

PROCEDURE DESCRIPTION:

1. Generation of an AVS vector 4 bytes uniform field, containing the input image:

- 1.1 Generate an AVS field header specifying the input image as containing the data (see AVS documentation)
- 1.2 Inside the AVS Network Editor, Generate the following network:
 - . READ_FIELD (specify the file generated at 1.1)
 - . DOWNSIZE (to match the DEM resolution)
 - . EXTRACT SCALAR (3 instances one for each color, if color)

- image)
 - . COMBINE SCALAR (if BW image, connect the 4 COMBINE SCALAR inputs to the READ FIELD output, else connect each EXTRACT SCALAR output to one of the COMBINE SCALAR input)
 - . WRITE FIELD (specify the output file name)
- 2. Generation of an AVS scalar, integer or byte, uniform field, containing the DEM:
 - 2.1 Generate an AVS field header specifying the DEM as containing the data (see AVS documentation)
 - 2.2 Inside the AVS Network Editor, Generate the following network:
 - . READ_FIELD (specify the file generated at 2.1)
 - . FIELD TO INTEGER (if the input DEM is not already in integer format)
 - . WRITE FIELD (specify the output file name)
- 3. run DEM_TRI with the AVS field containing the DEM as input to generate the low resolution triangle mesh
- 4. run DEM_TRI_HIGH with the 2 AVS fields as inputs to generate the high resolution triangle mesh
- 5. modify the 2 open statements in the jupiter:/u0/home/dore/variable/geom_speak function to contain the low and high resolution triangle meshes files generated in 3 and 4 (if possible use relative paths for the files)
- 6. Generate a new version of the RUNME file, using the UNIX utility: Make (the makefile file is provided)
- 7. Copy the RUNME to a directory consistent with the files descriptions in the geom_speak open statements
- 8. run RUNME with the option -demo 1 and if the stereo server is activated add the -stereo option

3.2. PAIR OF NHAP ORTHO IMAGE GENERATION

INPUT DATA:

- a pair of scanned NHAP photographs
- Digital Elevation Model
- Ground Control Points

REQUIREMENTS:

- The NHAP images scanning must be performed in such a way that no geometric processing will be required for stereo visualization
- The scanned NHAP images must include the fiducial marks.
- The Ground Control Points must be in the same reference system as the DEM
- The Ground Control Points reference system must be supported by PHOTOG
- A minimum of 6 Ground Control points is required (T.B.C). These GCP must also be identified in both images
- LT and RAS2UT must be available
- The DEM must be in a format compatible with the INTERP_DEM module input definition
- The Vexcel SSCS resampling procedure modules must be available

PROCEDURE DESCRIPTION:

1. Format the NHAP images to the UTIFF format
2. Using LT, acquire: data required by PHOTOG (fiducial points, control points, ...), GCP. The GCP ground coordinates may come from different sources: MAPS (1:24000 or larger is recommended), GPS, ...
3. Format images coordinates and ground coordinates according to the PHOTOG input format (see Vexcel PHOTOG documentation)
4. Run PHOTOG
5. Run INTERP_DEM. At this stage, the user defines the resampling grid
6. Run SIMCAM to compute the resampling grid nodes image coordinates
7. Run GENERATE_RESAMP_DATA (in VAX::_DUA2:[EOS.NHAP.MISC]) to generate the inputs to the SSCS resampling procedure
8. Apply the SSCS resampling procedure

3.3. PAIR OF SPOT MULTISPECTRAL ORTHO IMAGES GENERATION

INPUT DATA:

- 2 MULTISPECTRAL SPOT level 1A tapes
- Digital Elevation Model
- Ground Control Points

REQUIREMENTS:

- The tapes must be in the SPOT Image Corporation format
- The DEM must have been preprocessed to be compatible with the GENERATE_RESAMP_DATA_DEM module
- The DEM and GCP must be in the UTM cartographic reference system associated with the NAD 27 datum
- no geoid correction (was not tested)

PROCEDURE DESCRIPTION:

1. Load the images
2. Extract the images from the image file, using READ_CHANNEL
3. Extract the ancillary data from the leader file, using READ_LEADER
4. Acquire the GCP, using LT
5. If required, preprocess the attitude data, using ATTITUDE
6. If required, preprocess the ephemeris data, using ORBIT
7. Format the ground and image coordinates to the SPOTCHECK format, using PRESPT
8. Estimate geometric parameters corrections, using SPT
9. Estimate the mapping from the images to the ground, using SAT_UTM
10. Estimate the mapping from the ground to the images, using UTM_SAT
11. Compute the inputs to the SSCS resampling procedure, using GENERATE_RESAMP_DATA_DEM
12. Apply the SSCS resampling procedure

3.4. AIF IMAGE CUBE GENERATION

INPUT DATA:

- variable number of raster images

REQUIREMENTS:

- the images should all have the same size
- the images should be 8 bits per pixel
- an aif image is available

PROCEDURE DESCRIPTION:

1. Extract the aif header from the aif image, using for example EXTVOL. Do not forget to include the double ^L
2. Edit the aif header. More precisely, update the following fields:
 - width, should be image width
 - height, should be image height
 - depth, should be number of layers
 - pixel, should be 'p8' (8 bits per layer)
 - encoding, should be raw (no encoding involved)
3. Iteratively, append each layer to the file, composed of the header and the previous layers

3.5. RASTERIZE A DLG SET

INPUT DATA:

- USGS 1:100,000 DLG files

REQUIREMENTS:

- One disk file per tape file

PROCEDURE DESCRIPTION:

1. Extract the lines points planimetric coordinates, using READ_LINE
2. Write a script file, which at first, generate the output image with one of the processed DLG file, using RASTER_DLG, then iteratively add each DLG file, using RASTER_DLG

3.6. PROJECT DLG POINTS INTO SPOT IMAGES

INPUT DATA:

- USGS 1:100,000 DLG files
- MULTISPECTRAL SPOT images
- DEM

REQUIREMENTS:

- One disk file per DLG file
- the SPOT images must have been processed by SPOTCHECK
- the DEM must be in UTM, using NAD 27 datum
- the DEM must be in a format compatible with APPLY_UTM_SPOT and INTERP_HEIGHT_LINE

PROCEDURE DESCRIPTION:

1. Extract the DLG lines points planimetric coordinates, using READ_LINE
2. Compute the DLG lines points heights, using INTERP_HEIGHT_LINE
3. optionally, filter the lines, using FILTER_XYZ and REM_PAIR_XYZ
4. Compute the DLG lines SPOT images coordinates, using APPLY_UTM_SPOT

3.7. PROJECT DLG POINTS INTO NHAP IMAGES

INPUT DATA:

- USGS 1:100,000 DLG files
- Scanned NHAP images
- DEM

REQUIREMENTS:

- One disk file per DLG file
- the NHAP images must have been processed by PHOTOG
- the DEM must be in UTM, using NAD 27 datum
- the DEM must be in a format compatible with INTERP_HEIGHT_LINE

PROCEDURE DESCRIPTION:

1. Extract the DLG lines points planimetric coordinates, using READ_LINE
2. Compute the DLG lines points heights, using INTERP_HEIGHT_LINE
3. optionally, filter the lines, using FILTER_XYZ and REM_PAIR_XYZ
4. Compute the DLG lines NHAP images coordinates, using SIMCAM

3.8. GENERATE ICE MOVEMENT

INPUT DATA:

- 2 ERS1 image files (provided by GPS)
- motion vector file (generated with the 2 ERS1 images)

REQUIREMENTS:

- The images origin SSMI coordinates and pixel size must be available in an ASCII file (same file name with a ".def" extension)
- The motion vector file first record must be removed
- The images must be stripped of their headers



PROCEDURE DESCRIPTION:

1. From their geographical coordinates, generate the motion vectors target image coordinates.
2. From their geographical coordinates, generate the motion vectors source image coordinates, corresponding to a fraction of the displacement.
3. For each displacement fraction, generate the line and pixel files, representing the mapping from the target image to the partially displaced source image.
4. Generate the source image resampling grid.
5. Resample the source image.
6. Update the image cube.

ATTACHMENT E

ANALYSIS OF SPECIAL PURPOSE SOFTWARE SOFTWARE “SPECTRAN”

Vexcel Corporation

EOS WORKSTATION

SPECTRAN

Spectral Analysis Prototype

April 8, 1991

by

K. Maurice

INTRODUCTION

This document describes the EOS Prototype Image Analysis Workstation (EOSW) effort for the spectral analysis component of the work done. This component represents an exploration of the capability required by users of the workstation to analyze multi-dimensional images.

The workstation specification places spectral analysis capability within the context of image cube analysis functionality. Details of the work performed for this area are described in the following sections.

IMAGE CUBE ANALYSIS AND SPECTRAL SIGNATURES

In general, the workstation is to provide the user with an assortment of image cube analysis functions including visualization, feature acquisition, signature analysis, GIS interfacing and data selection. The work described here focuses on signature analysis but also indirectly addresses other elements as well, including image cube visualization.

The idea behind signature analysis is to allow users to examine and acquire signatures from a single sensor and make comparisons with previously acquired signatures or other externally introduced signatures. In particular, the ability to analyze multispectral radar or optical signatures is required. The user must be able to explore an image cube of homogenous layers, viewing individual pixel responses or functions of spatial area responses (ie. region of interest averages). The user must also be able to save signatures in a database and to select from this database for comparison with other image data signatures. Comparison with signatures obtained outside of the system (ie. from a laboratory) must also be supported. Reference to planimetric data (thematic or image) must be provided.

A complete functionality is described in the EOSW specification documents *Functional Description* and *Environmental and Behavioral Models*. Related work in the general area of image cube analysis is described in Millot, et. al 1991.

SIGNATURE ANALYSIS PROTOTYPE

To test the concepts of signature analysis a demonstration prototype was developed to evaluate a number of user scenarios on sample application test data. This prototype incorporated basic functionality as outlined in the specification including:

- image pixel signature acquisition and display,
- input selection and display of laboratory signatures together with image signatures,
- spatial band and spectral band interval selection and display,
- polygonal region of interest signature average and display, and
- image-cube based display where reference cube images are compared with image cubes.

Additionally, a 3D-cursor style of roaming was implemented that maintained a cursor through both a multi-band cube and a reference cube of derived products and provided visualization of spectral and spatial views.

The prototype was required to provide a minimal functional equivalent to existing industry systems such as JPL's SPAM/ISIS (Mazer, et. al 1988, Torson, 1989), while at the same time fitting into the EOSW framework.

COMPUTER CONFIGURATION AND PERFORMANCE

The demonstration prototype was implemented using Precision Visual Inc.'s PV-Wave procedures. In principle it can operate off of any sized cube, utilizing a cache for cubes bigger than available or specified memory. The demonstration used a data set size that could fit entirely into memory at once and still provide reasonable performance on our platform (Sun 4/280). An increase in memory and/or CPU speed is directly reflected in the performance of the program as evidenced by a marked improvement on a Sun Sparcstation 4/370 - even when output was generated across a network (via the X Window system).

The total memory used by the various input data totalled less than 17MB. This was a size that was on the Sun 4 tolerable but on the Sparc 4 much better. The result of performance loss is seen in the slow response of the interactive display updates to the free-hand movements of the mouse. A very fast computer would more closely approximate an instantaneous pixel response display.

DESCRIPTION OF INPUT DATA USED

Use of any type of image cube is possible but of most interest are those with multiple channels from a continuous range of like-unit values. For this prototype we used an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) data cube (210 bands x 512 lines x 614 samples) and a cube derived from it, as inputs, as well as a set of laboratory signatures provided by Fred Kruse at the Center for the Study of Earth from Space (CSES), University of Colorado. The AVIRIS cube was taken over the Northern Grapevine Mts. of Nevada on May 5, 1989. The cube was represented in raw JPL-calibrated form (with spectral resampling) for spatial band displays and in CSES-calibrated/normalized fashion for spectral displays and pixel signature plots. The spectral data were calibrated to field reflectance using the "empirical line method" (see Roberts, 1985) by Fred Kruse at CSES and normalized to a continuum-removed spectrum (see Kruse, 1990) to allow direct signature and laboratory comparisons. Derived types included two principle component bands and a classification image and reflect the ability to use registered feature or GIS images of thematic nature.

All cubes were scaled from 12-bit values to 8-bit bytes (0-255) and a subset spatial area was chosen (lines 10-209 and pixels 300-499). The spatial area was selected for its mineral content, including dolomite, calcite, hematite and goethite. All 210 spectral bands were used. The data cube used was thus 200 lines x 200 pixels x 210 bands for 8.4 MB. A spatial and spectral cube was needed each of this size as well as a reference cube 200x200 by 3 bands. The total memory used was therefore $2 \times 8.4 + 120K \sim 17$ MB.

SETTING UP AND RUNNING SPECTRAN

The Wave environment must be set by "sourcing" the file `$WAVE_DIR/bin/wvsetup`. Then the script `runspectran` can be run to start up the demonstration program. A menu choice between

running the program and exiting must be selected. Upon selection of running the program the reference window-state is entered. When the states are exited the user must once again make the menu choice - this time to choose quit to exit altogether.

DEMONSTRATION USER INTERFACE

The basic flow of control through the demonstration program is that of a set of sequential states driven by the left mouse button connected to the user's console. The system starts in State₀ and with each click of the left mouse button proceeds to the next state State_{*i*} until it reaches State_{*n*-1} at which point the next left mouse click brings the user back to State₀. A press of the right mouse button from any state causes the entire loop to exit. The middle button is used for state-specific functions. The states are:

- spatial reference
- row spectral image
- column spectral image
- spatial image
- prototype signature selection
- signature-based image band selection
- reference-based region-of-interest average

A particular window is associated with each state although actions within a given state often affect other windows. Emphasis was placed on the linking of windows and their actions.

The movement of the current state's cursor (if it has one) causes windows with corresponding cursors to update their own cursor. A cursor is plotted as the intersection of two lines centered at a point and extending across both image directions. The net result is a near-instantaneous multiple slice plane view of any point in the cube and its reference (spatial view only). Figure 1 is an example showing the overall program display while within the Signature-based Band Selection Window-state as described below.

Each window-state is described below in terms of:



- its own actions and the effects on its own and other windows and
- the effects of other windows' actions on it.

Throughout the description of states the following convention is assumed:

<c1, c2> refers to a window image's x or pixel location (c1) and y or line location (c2).

Also, the plot window will always contain three signatures for any given state: the current pixel location $\langle x, y \rangle$, the current prototype substance and the last region-of-interest average, all from s_1 to s_2 .

REFERENCE WINDOW-STATE

This is the initial state that the program starts up in. In this state the user can move the cursor via the mouse to any $\langle x, y \rangle$ pixel in the current reference cube layer and view the corresponding spectral response and input image cube slices in the other windows. This state is initialized at startup as described in the following.

INITIAL VALUES OF THE REFERENCE STATE

The value, r_z , for the current reference cube layer and the coordinates x, y, z for the current image cube point are initialized. The current reference cube layer R_{r_z} is displayed. The reference cube cursor is set to $P_{\langle x, y \rangle}$ and displayed on the reference layer. The current image cube band is set to z and the corresponding layer is displayed in the spatial image cube window. The spatial image band cursor is also set to $P_{\langle x, y \rangle}$ and displayed over image cube band I_z so that point P is identified on both the reference and input image. The current laboratory prototype L is set to L_j . The spectral display range is set to be from s_1 to s_2 and this range is displayed in the plot (spectrum) window for the laboratory prototype L_j . The response for the current pixel P is also displayed in this window from s_1 to s_2 . The region of interest average spectrum is initialized to a constant value A and displayed also in this range.

In our example, r_z is set to 1 so that the initial reference layer corresponds to the first layer in the

reference cube. This layer contains a principle components image and it is displayed with its cursor set to $P_{\langle 100,100 \rangle}$, corresponding to the center of the image. z is set to 52, corresponding to AVIRIS band #52 of the input image cube, centered on $1.056 \mu\text{m}$, and it is displayed with its cursor at P . The row spectral plane is set to y , where y is the line number of P and the spectral response of line y for all pixels and bands of the image cube is displayed. The row spectral plane cursor is set to $\langle x,z \rangle$ where x is the pixel number of P . The column spectral plane is set to x , where x is the pixel number of P and the spectral response of pixel x for all lines and bands of the image cube is displayed. The column spectral plane cursor is set to $\langle z,y \rangle$, where y is the line number of P . The spectral range for the signature window is set to be 0...209 which is the full range for the AVIRIS cube. The current laboratory prototype is initialized to be Dolomite. The region of interest is set to $A = 0$.

REFERENCE WINDOW ACTIONS AND EFFECT ON OTHER STATES

When the user moves to pixel location $P_{\langle i,j \rangle}$ in the reference cube image, the spatial image window cursor will also move to location $P_{\langle i,j \rangle}$ on the current image cube band. The row spectral image window will at that time display the spectral responses of line j and its cursor will be updated to location $\langle i, z \rangle$. The column spectral image window will display the spectral response of column i and its cursor will be updated to location $\langle z, j \rangle$. While in State_0 , cursor movements in the reference window will only cause cursor movements in the x component of the row spectral plane since the image band is fixed. Similarly only y movements will be reflected in the column spectral window. The current pixel signature is plotted in the plot window for each new mouse movement to location $P_{\langle i,j \rangle}$ with range $s1$ - $s2$.

The values for x,y,z and rz are retained from State_{n-1} when the reference state is once again reached via the left mouse button and the behavior is as described above.

REFERENCE WINDOW FUNCTION

When in the reference state the user can flip through the reference cube bands by pressing the middle mouse button. Successive bands are displayed until the last is reached at which point the

sequence wraps around to the first image. The cursor and other windows are maintained as discribed above.

ROW SPECTRAL IMAGE WINDOW-STATE

This state follows the reference state. In this state the user can move the cursor via the mouse to any $\langle x, z \rangle$ pixel in the current row spectral cube layer and view the corresponding spectral response and image cube slices in the other windows. This state (like all others) is initialized to have x, y, z , and rz from the previous state.

ROW SPECTRAL WINDOW ACTIONS AND EFFECT ON OTHER STATES

When the user moves to pixel location $R_{\langle i, z \rangle}$ in the row spectral cube image, the reference window cursor will move to location $P_{\langle i, j \rangle}$ on the current reference cube band R_{rz} , with its cursor fixed at line j . The spatial image window will at that time display the image I_z and its cursor will be updated to location $\langle i, j \rangle$, where j is again fixed . The column spectral image window will display the spectral response of pixel i and its cursor will be updated to location $\langle z, j \rangle$, where only z may have changed. The current pixel plotted in the plot window will vary with each new mouse movement where there is a change in i , the pixel number. The range is $s1-s2$.

ROW SPECTRAL WINDOW FUNCTION

The lower plot limit, $s1$ may be defined by moving the cursor to a desired z coordinate and pressing the middle mouse button. The second (required to be greater) limit may be defined by moving to a second z coordinate and pressing the middle mouse button again. The plot window will be immediately updated to reflect the new range. Figure 2 shows the row spectral window along with the reference window.

COLUMN SPECTRAL IMAGE WINDOW-STATE

This state is identical to that for the row spectral state except that here the pixel number is displayed through all lines and bands. The spatial plane is updated to reflect changes in z.

This state was, in fact, left unimplemented for the present effort due to time and performance constraints. It would, in theory, also have the middle-button range selection function also.

SPATIAL IMAGE WINDOW-STATE

This state is identical to the reference state in that cursor movements cause corresponding effects on spatial and spectral displays and cursor locations. There is no middle-button function, however. In the reference state, mouse movements caused the spatial image window cursor to change. In this state, similarly, mouse movements update the reference state's cursor.

PROTOTYPE SIGNATURE WINDOW-STATE

In this state there is no roaming, only a middle-button function. Each press of the middle mouse button causes the current prototype to be set to the next prototype in the database and then displayed. Sequencing continues until the last element, at which point the next press selects the first element in the list again. The following substances were obtained from CSES and used in the demonstration prototype:

*actinol, alunite, buddingtonite, calcite, chlorite, dolomite, drygrass, elvimg.,
epidote, goethite, gypsum, hematite, illite, kaolinite, montmorillonite
pyrophy, tremolite*

SIGNATURE-BASED BAND SELECTION STATE

This state allows the user to view the response, over all bands, of the individual pixel defined by $P\langle x,y \rangle$, as an image of 1 dimension by $\max(Z)$ dimension (eg. 210 in our case). A small window, 210 pixels by 1 is displayed and the user may point the mouse into this pixel array and press the middle mouse button to highlight a particular band layer. When a pixel is highlighted a



vertical line will appear in the plot window exactly on the band #/wavelength of the plot graph. At the same time the spatial image window will be replaced with the image of the spectral band selected on the pixel array. The spatial image cursor temporarily disappears.

In this way, the user can examine the spectral responses of various spatial pixels and then, according to spectral features noticed in the plot window, selectively view the corresponding image planes.

REFERENCE-BASED REGION-OF-INTEREST STATE

In this state the reference window again is used, this time to draw a polygon. The user moves the mouse to a desired point and presses the left mouse button to start the polygon. Successive presses of this button connect lines until the right button is hit at which point the polygon is automatically closed. The average signature for this area is computed and displayed in the plot window from s1 to s2. A press of the first button puts the user into the next state (the reference state). All polygons are plotted and remain on both the spatial and reference images until the program exits. Figure 3 shows a polygon after it has been defined on the reference window.

SIGNATURE PLOT WINDOW

The plot window is not a state but is an effect, "only", of other window-states and reflects the *current pixel*, *prototype substance*, and *region-of-interest* responses. This window is used to compare these three signature types for various ranges (s1 to s2).

USE OF X WINDOW MANAGERS

A number of window manager functions can be performed. These include moving, iconifying, and reordering windows. Thus it is possible to arrange windows in any fashion and to have a number of other applications running with the demo program (if there is memory).

APPLICATIONS USER SCENARIOS

EXPLORATION BY SUBSTANCE TYPE

In this case an investigator focuses on a substance and is trying to find out where this substance is found in the imagery. Pixels can be examined until a match is visually obtained.

EXPLORATION BY FEATURE AREA

In this instance, the user is interested in what the signatures are around a given area or at a given feature. The user can point to a feature or area and examine the corresponding signatures or signature average.

ADDITIONAL APPLICATIONS

A residual cube was used, representing a product derived from the test AVIRIS image cube. In this case the differences (residuals) between a laboratory prototype and a cube were used in combination with the reference cube (Figure 4). See (Maurice, Kober and Kruse, 1991) for details of visualization of these residual cubes. In general the tool can be useful for examining a wide variety of related image cubes.

ADVANTAGES OF A FULL SYSTEM IMPLEMENTATION

In an actual implementation there would not be a procedural or state-based interface. When the user moves into a certain window the system would recognize this and be then driven by this window. A general spatial and spectral selection and scaling mechanism would be in place also. Automated techniques could also be used to drive the display of pixel responses (eg. classification).

A real system would integrate spectral analysis functions so that they would become part of a much larger set of overall functions. It would be possible to use other EOSW sub-systems in conjunction with spectral signature analysis and also fully utilize feature capabilities and

database management. All of the capabilities of the prototype demonstration plus much more would be available with a full system implementation, and designed in a way that allowed an open-ended interface for connecting a wide variety of functions into specific applications.

REFERENCES

1. Millot, M., Maurice, K., Karspeck, M., Leberl, F. (1991) "Generalization, Visualization and Analysis of Image Cubes".
2. Mazer, A.S., Martin, Miki, Lee, Meemong, Solomon, J.E., (1988), "Image processing software for imaging spectrometry data analysis": Remote Sensing of Environment, v.24, n0.1
3. Torson, J.M., (1989), "Interactive Image Cube Visualization and Analysis": Chapel Hill Workshop on Volume Visualization, May 18-19, 1989, Conference Proceedings.
4. Roberts, D.A., Yamaguchi, Y, Lyon, J.P., (1985), "Calibration of Airborne Imaging Spectrometer Data to percent reflectance using field spectral measurements": Proceeding, 19th International Symposium on Remote Sensing of Environment, Ann Arbor, MI, 21-25, Oct. 1985.
5. Kruse, F., Seznec, O., Krotkov, P., (1990), "An expert system for geologic mapping with imaging spectrometers", SPIE Vol. 1293 Applications of Artificial Intelligence VIII
6. Maurice, K., Kober, W., Kruse, F. (1991), "Visualization Environment for Imaging Spectrometry", 8th Thematic Conference on Geological Remote Sensing, Denver, CO. April 29-May 1, 1991.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

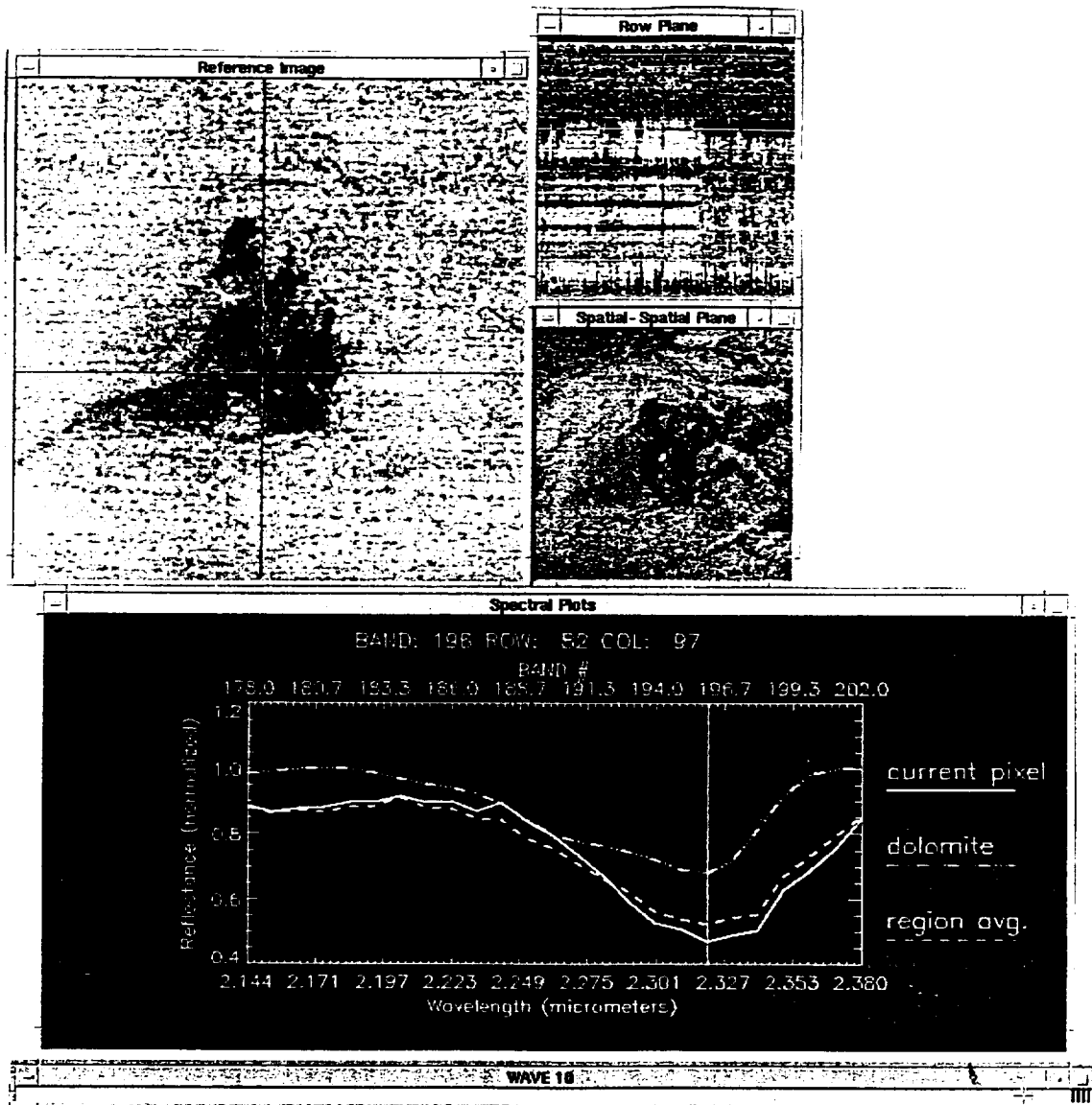


Figure 1 - Signature-based Band Selection, band 196, line 82, pixel 97

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

2288

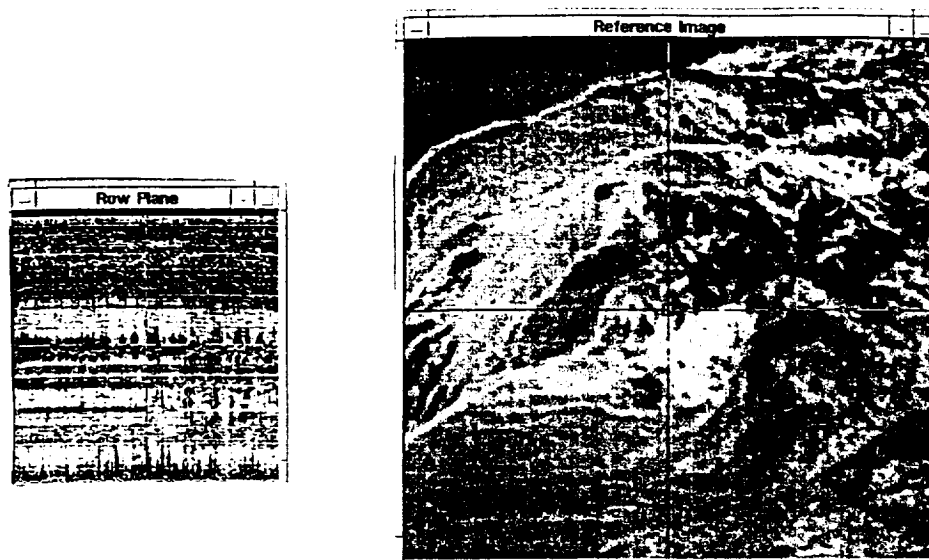


Figure 2 - Row Spectral State, with $z=198$, $rz=1$, $x=99$, $y=95$



Figure 3 - Region-of-interest State, with a defined polygon on the Reference Image

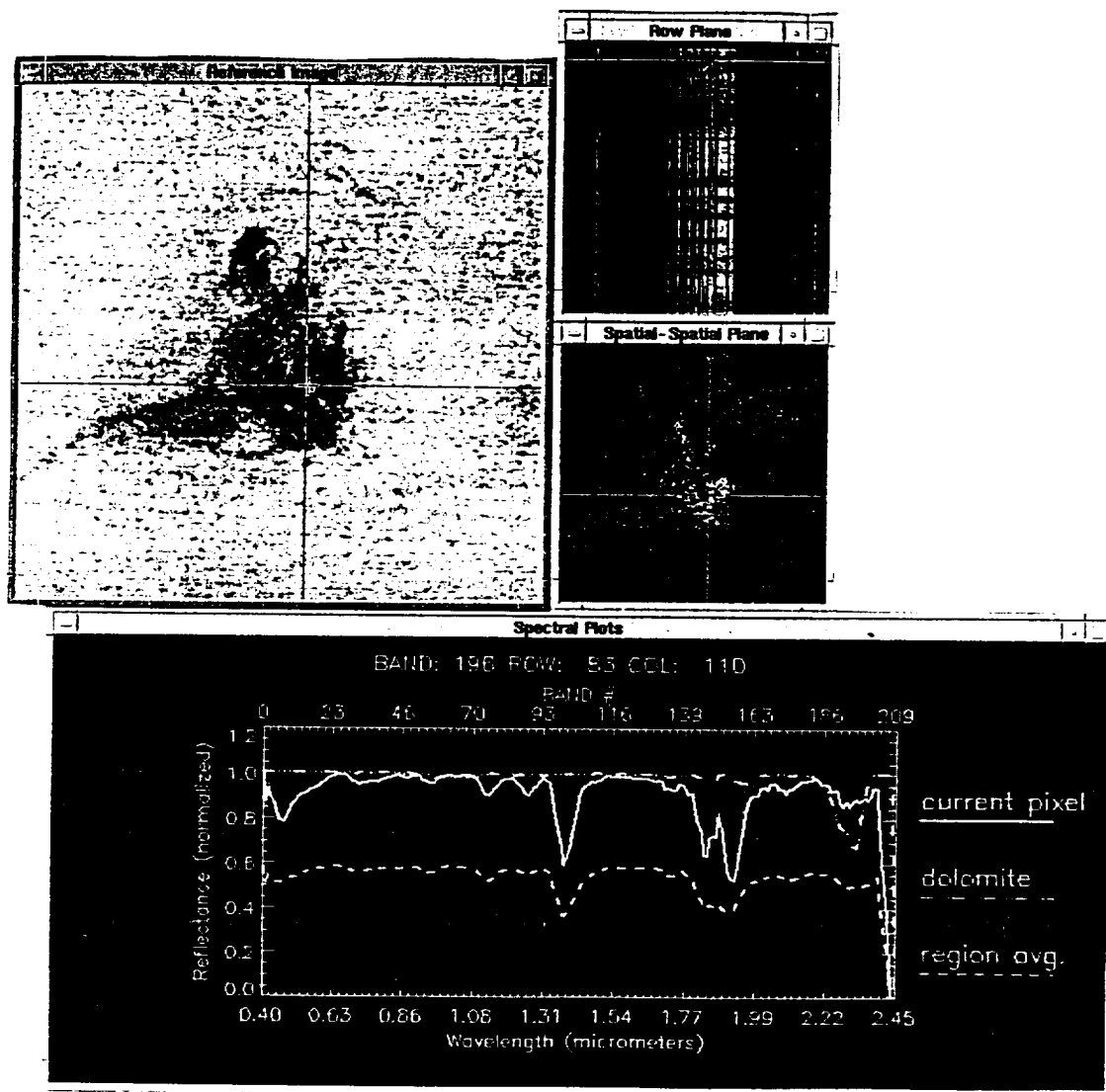


Figure 4 - Reference State, with a Dolomite residual cube compared with PCA #2

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

ATTACHMENT F

ANALYSIS OF THE THIRD PARTY SOFTWARE “GRASS”

- **Investigation**
- **Demonstration Description**



Vexcel Corporation

EOS Workstation

REPORT:

**Investigation
of**

United States Army

Construction Engineering Research Laboratory's

**Geographical Resources Analysis Support System
(GRASS)**

prepared
by

Kelly E. Maurice

September 16, 1990



Introduction

The purpose of this document is to present an evaluation of the GIS package GRASS with emphasis on GRASS functionality useful to the EOS Workstation.

Additionally, a brief analysis, on various levels, of the approximate work required to implement an interface to GRASS is done. These levels range from hi-level to program and library level.

Finally a set of prototype scenarios is described for a sample data set produced from another project at Vexcel. These scenarios are the basis for a demo encapsulating the GRASS investigation activities and running currently on the Sparcstation 4. The demo/prototype effort is meant to illustrate, using the actual GRASS software, the utility of GRASS for a typical EOSW or other similar project and highlight the characteristics unique to a GIS. The demo itself is described in the document "EOSW Demo of GRASS". This document addresses issues relating to the data format conversion and input loading of the demo data sets since this activity is in fact part of one of the interface levels described.

The *overall* objective of the GRASS investigation activity has been familiarization with GRASS functionalities for the following future possibilities:

- 1) porting of X GRASS to the Stardent; an X version is expected by 1990-1991;
- 2) running GRASS from the Stardent using file transfer methods;

(See the VEXCEL Memo 10 August 1990 from Karspec and Leberl). Another possibility would be to directly interface with GRASS libraries or in fact enhance those libraries. A more in-depth description of the EOSW GIS requirements can be found in the EOSW Analysis and Functional Specification documents. The next section describes the structure of this document and primary topics covered.

Document Organization

This document is aimed at outlining the basic capabilities of the GIS, GRASS, developed at the United States Army Construction Engineering Laboratory (USA CERL). These capabilities are evaluated within the context of Vexcel's EOS Workstation requirements, some of which include GIS support. A prototype/demo set is described to show some of the possibilities available with GIS use.

The first section provides background on GIS technology in general and GRASS functionality and EOSW GIS requirements in particular. A brief summary of the current Vexcel GRASS installation is given.

Next a detailed description of GRASS functionality useful to the EOSW is presented. This is described for functions required by the workstation as well as those not required but possibly desirable. With the new man-power estimates and projected work for EOSW, there is the possibility for making up for lost functions by interfacing to a GIS or other system which already contains such functions.

The possible interface levels to GRASS are described. These include *basic*, *shell*, *program*, and *integrated*.

EOS Workstation G.R.A.S.S. Investigation Report

Lastly, we provide a description of the prototype scenarios with emphasis on the data sets themselves, data set preprocessing, importing of the sets to GRASS, and GRASS operations performed to generate and visualize the data sets. No attempt has been made to describe the exporting of data from GRASS back into another Vexcel (or other) system. It is assumed that this can be done and this is in fact an important path to be provided to EOSW users (See EOSW Analysis).

Background

GIS Technology

"GIS technology" is a fairly recent phenomena, beginning in the 1960's (Smith, et. al, 1987). A GIS is to consist of five component sub-systems (Knapp, 1978):

1. data encoding and input processing
2. data management
3. data retrieval
4. data manipulation and analysis
5. data display.

In addition a contemporary GIS must address these additional requirements (Smith, et. al., 1987):

- a. ability to handle large, multi-layered, heterogenous databases of spatially indexed data;
- b. ability to query such data bases about the existence, location and properties of a wide range of spatial objects;
- c. an efficiency in handling such queries that permits the system to be interactive;
- d. a flexibility in configuring the system that is sufficient to permit the system to be easily tailored to accomodate a variety of specific applications and users;
- e. an ability of the system to 'learn' in a significant way about spatial objects in its knowledge and databases during use of the system.

It is believed that to satisfy the above requirements a traditional GIS must now also incorporate into its development the following issues:

- a. software engineering
- b. spatial data models and data structure
- c. vector models
- d. Tessellation models
- e. RDBMS
- f. algorithmic considerations
- g. knowledge-based approach
- h. environment integration

The essential trends appearing to drive GIS development today are:

- . high rate of generation of spatially indexed data from a variety of sources
- . the demand for GIS to handle such volumes in a large variety of decision-making

EOS Workstation G.R.A.S.S. Investigation Report

situations is increasing dramatically.

Typically there is seen to be a large overlap of GIS with a number of fields:

- . remote sensing
- . image processing
- . computer graphics
- . database management
- . CAD
- . cartography and mapping
- . (photogrammetry)

(Goodchild, 1987) notes "...the ability to manipulate spatial data into different forms and to extract additional meaning from them is at the root of GIS technology."

EOS Workstation GIS Requirements

The desirability of a GIS for the EOSW effort fits with the inter-disciplinary nature of GIS. The EOSW can be a tool for a wide number of applications, incorporating many of the fields listed above. It seems natural for such a system, which inherently must deal with large amounts of heterogeneous spatial and non-spatial data to use a GIS. EOSW GIS requirements in general demand that user be able to:

- a. import data from a GIS base into the workstation for further processing
- b. export data from the workstation to a GIS for further GIS investigation.

In the original workstation analysis GIS interaction was to be supported at both the image cube level and at the image feature level. Thus GIS features could be imported to and exported from the workstation through image cubes as well as image features. Further, in the updated analysis for *prototype signatures* it was discovered that band-reduction could be done by feature area.

As of this writing GIS feature interaction is to be limited to simple control points since the workstation will deal only with such features and not at first operate off of polygon-based features. We are therefore investigating the use of a GIS in defining and editing features that would once have been done in the workstation.

A second, more fundamental use of a GIS under the current scope would be to just add a GIS as a basic database/data source engine. That is, a GIS could be used to, for instance, label classes automatically identified in a classification scheme and to then export them into the workstation as image cube layers.

In either case, depending on the level of GRASS functionality retained in the particular interface approach taken, it may be possible to recover *unimplemented* EOSW functionality (eg. feature editing, spectral analysis, etc.). Recovery of functions is discussed in the section describing GRASS functionalities not included in the current EOSW specification.

The requirements at this time for the level of GIS "integration" are less firm. Later sections in this document address the various interface levels possible for GRASS.

The GRASS System

EOS Workstation G.R.A.S.S. Investigation Report

The following characterizes the GRASS system:

- . **GIS** - it is a traditional GIS (see above section on GIS technology)
- . **Raster-based** - all data processing is cell-based and not vector-based with the exception of digitizing and graphics overlay
- . **tool set** - GRASS consists of various tools that can be used among themselves or with other systems
- . **I/O system** - provides mechanisms for input and output from and to external environments

GRASS (3.0) consists of roughly 200 different programs which are accessible through menus or interactive or non-interactive keyboard commands. The basic GRASS functions can be broken up into the following four categories:

- 1.) **GEOGRAPHICAL ANALYSIS (GA)** - map analysis and overlay capabilities including proximity analysis, logical reasoning, weighted overlays and neighborhood processing are brought to bear on various derived data to answer land-use and other questions.
- 2.) **IMAGE PROCESSING (IP)** - images can be geocoded and classified; results can be saved in a database for later combination with elevation, slope, geological or other map data.
- 3.) **MAP DISPLAY (MD)** - monitor and hardcopy output displays in various size formats.
- 4.) **DATA INPUT (DI)** - data capture including tape input, digitization from paper maps or digital imagery, vector (DLG, ascii) and elevation (DEM, DTED) input and generic cell and vector input.

All processing is with reference to a unique geographical *location*; a *mapset* is the next unit of data partitioning; finally there are cell maps, vector files, window files and a number of other auxiliary and support files associated with the average GRASS application. Devices supported by GRASS include a digitizer, hardcopy output and graphics monitor. These can be configured for various commercial products. There is on-line help and good documentation for all functions. GRASS is an evolving product and it is expected to grow and change in the future.

Installation Notes

Vexcel has purchased GRASS version 3.1 from DBA systems and has installed it on the Sparcstation 4 (node "Vexcel"); the installation was performed using the installation guide and the script GISGEN. No problems were encountered. No digitizer has been installed nor any hardcopy output device. The current graphics device is the Sunview console.

NOTE: we have found through experimentation that the best way to use GRASS is to have one terminal (ASCII) dedicated to program interaction and one monitor (the graphics sunview console) for display. Having both in one window tends to cause the system to lock up when using certain functions. It is not clear whether this is a bug in GRASS or due to the fact that we are

EOS Workstation G.R.A.S.S. Investigation Report

running on a Sparcstation under SUNOS V3.01. This situation will go away with an X Window version of GRASS! The basic environment without source code and with prototype files occupies roughly 70M bytes of disk space.

GRASS Capabilities

General System Structure

Most basic GRASS functions can be accessed in two ways: through a hi-level menuing program called "grass3" or through a command line interface invoked as "GRASS3". The command-line interface is further broken up into interactive and non-interactive programs. Not all of the functions in the command-line version are available in the menu version. The menu version has been found to be of *limited* usefulness except to novice users. The entire user interface will probably have a facelift for the X window version where graphics interaction (mouse movements and display windows) will be merged with the text prompts and user interaction (commands, parameters). From here on, user interface and function discussions will all refer to the command-line version of GRASS (GRASS3).

The functions themselves are broken up into five groups:

- . Map development programs
- . Interactive display/analysis programs
- . Non-interactive analysis programs
- . Non-interactive display programs
- . Image analysis programs

The use of each of these programs depends on the level of interface used for GRASS. There is overlap between some non-interactive and interactive programs: the interactive programs call the non-interactive programs, providing a friendly interface to the user who does not then have to formulate complex command-line statements. In the hi-level EOSW interface with GRASS the system would be used as is and we would directly use the various commands. At the programming level we must address the issues involved with using GRASS library functions. At the lowest level we must be concerned with decoupling the core GRASS functions from the rest of the system overhead functions (ie. user-interface, etc.). An appendix included as part of this report lists a brief description of all of the GRASS functions (both interactive and non-interactive). Refer to the GRASS User's Manual for complete function information and tutorials.

Primary GRASS Functions of Interest to EOSW

Functions of primary interest to the EOSW come from the basic groups. These functions fall into the class of *required GIS functions* - those that **must** be supported by the EOSW. These are in contrast to those functions that were initially targetted but have now been deferred due to manpower constraints. Note that these are somewhat undefined requirements and have at this time assumed the status of functions that are "unique to GIS" and "needed by the EOSW". Therefore this class of functions precedes all others - that is, they must be functions that the EOSW does not have and that in fact justify the very existence of a GIS as part of the EOSW!

Although this functionality is not specified it is anticipated that the baseline GIS functions that would be supported would involve an interface to *elt*, the electronic light table. *elt* is responsible for collecting EOSW control points. It is expected that EOSW users will be able to additionally define area or line features within *elt* and that these features can be exported to a GIS

1

EOS Workstation G.R.A.S.S. Investigation Report

(eg. GRASS). These features in conjunction with images extracted from EOSW image cubes and thematic maps from sources such as the USGS would provide data for useful GRASS layers to be created. These, in combination with vectors (possibly modified or converted from DLG) could then be exported back to EOSW image cubes.

It is assumed that features output from *elt* would be of a simple point form; this data can be converted to GRASS ascii vector (digit) format using a program (in-house) like *tovect.c*. It is assumed that no topological information is included with this data.

The most useful functions for this kind of processing are:

MAP DEVELOPMENT (MD):

1. *import.to.vect* - provides basic import mechanism of simple ascii vector files to the GRASS vector format; also handles input of DLG files.
2. *support.vect* - this function extracts topological information from a file created by *import.to.vect*; this must be run before running *digit*.
3. *digit* - this general purpose digitizing program is useful for editing points and lines (delete, snap, label, add, etc.); in particular it allows the assignment of categories (attributes) to vector file features; categories must be assigned before creating a cell map (raster file) from any imported vector file; in addition to assisting the import process, *digit* also provides a number of functions for verifying data and for creating additional vector and/or cell files; it will allow overlay of 1 additional vector file and backdrop of an image file (cell) during editing; other capabilities include geographical area selection, zooming, text labelling, map digitization and a number of other functions.

DIGITAL ELEVATION DATA:

4. *Mimportcell* - imports an ascii file into a cellfile; this is GRASS's basic mechanism for importing raster files: DEMs, images, thematic maps, etc. A header is required for each input file describing geographical area, map name, etc.

INTERACTIVE ANALYSIS AND DISPLAY:

5. *cell.stats* - gives statistics about layers in terms of acres, hectares or sq. miles per category; this would allow quantitative determination of areas of polygons in terms of geographic units
6. *coin* - charts mutual (coincidence) of all categories from one cell map with another
7. *combine* - general-purpose map overlay tool supporting combinations of map categories from several maps using AND, OR, GROUP, NAME, COVER, OVER.
8. *copy* - copies existing database files (cells, vectors, etc)
9. *describe* - prints list (range) of values in a layer

EOS Workstation G.R.A.S.S. Investigation Report

10. *distance* - proximity analysis based on distances from selected categories
11. *mask* - generate/remove masks based on cell file categories
12. *neighbors* - enhance or subdue data values based on surroundings
13. *reclass* - recategorization of existing maps
14. *Gmapcalc* - the classic GIS function - a must for claims to GIS support

Note: there is a lower level interface for import.to.vect (a.b.vect plus build.vect) and other programs above but this issue is more one of interface level and will be discussed later.

Secondary GRASS Functions of Interest to EOSW

The following could also be important as baseline GIS functions: *watershed*, *grass.armsed*, *slope.aspect*, *Mll2u*, *Mll2g*, *grow*, *basins.fill*.

Possible GRASS Functions to Replace Lost EOSW non-GIS Functions

These functions are those that were required for the EOSW but are being eliminated due to cost overruns. These include:

1. *mapmask* - creates cookie-cutter masks of circles, polygons, etc. from the user; this would supply EOSW a feature editing tool capable of making masks.
2. *i.cluster* - generated spectral signatures for land cover types in an image using a clustering algorithm; this would supply a signature function.

Additional Functions of Possible Use to the EOSW

A number of functions may also be interesting to have: *clump*, *paint*, *grass.bnoise*, *patch*, *sites*, *i.maxlik*, *i.points*, *i.rectify*, *random*. See the User Manual for details of these and other functions.

Levels of EOSW-GRASS Interface

High (data format) Level

The hi-level interface would simply consist of a mechanism to import/export files to/from GRASS. Thus a program to convert EOSW files to GRASS import format would be required as well as a program (or module) to convert exported GRASS files back to EOSW format. The EOSW user then runs the GRASS system as is. This level would be very easy to implement and would also allow the full use of all GRASS functions. It is expected that *at least* this level of interface would be achieved. This is the level used for the prototype scenarios described in this document.

NOTE: we have not as yet found a way to tell GRASS to "show the category layer for a cell"

— — — — —

EOS Workstation G.R.A.S.S. Investigation Report

graphically; that is we would like to have a display "showing" what each category is with text. It seems that there **must** be a way to do this - we just have not found it yet!

Shell Level (loosely coupled)

This interface would use scripts to call the GRASS executables directly. Thus the GRASS system itself would not actually be modified but a lower level of control would be incorporated into the interface. Each GRASS function exists as an executable and a set of Unix-level scripts would be placed between the user and the programs. These scripts then would be supported by the workstation.

Program Level (linked libraries)

At the program level, we are actually **linking** into the GRASS system, using their object modules and libraries to build our own executables. These executables can then be hosted by the workstation. The GRASS system is divided up into 5 major libraries:

- . **segment** - paging and virtual memory
- . **GIS** - I/O for cell and support files
- . **Dig** - vector library database access
- . **Vask** - prompts, screen operations
- . **Graphics** - display and raster library

This level is described in the *GRASS Programmer's Manual*.

Integrated (source)

This level would be the most complex and involves integrating portions of the system into the EOS Workstation. Device and I/O operations would have to be decoupled from the basic functions. At this level we are taking over the software and in fact assuming responsibility for it. This would be a major undertaking. A more realistic and much simpler variant of this level would be to simply borrow and integrate selected source modules (ie. *Gmapcal*) that could be useful in their own right as stand-alone functions.

Prototype Scenarios

Motivation for Prototyping

The reasons for prototyping include the need to really "see" the benefits of using GIS support for the kinds of data that the workstation must handle. The first scenario presents a brief example of the sample database included with our GRASS version. The existence of a variety of features (roads, geology, land-use, land-cover, etc.) warrants inclusion since the second example uses a limited amount of category data. The last scenario is unimplemented (AVIRIS) but it is expected to be a useful example for the future.

Relationship to EOSW GRASS DEMO

EOS Workstation G.R.A.S.S. Investigation Report

The scenarios are in fact the demo items. This document goes into details of how the items were produced whereas the demo focuses on the results.

Level of GRASS Interfacing

The level is that of data format compatibility only (hi-level); this is the most straightforward and was judged sufficient for prototyping purposes.

Scenario 1 (GRASS sample set)

These sets are all provided by our GRASS distributor (DBA Systems) and describe a fictitious military camp scene in Spearfish S.D. USGS topographic maps are included for reference. No processing was done to create the original sets. Derived sets are produced using the GRASS Problem-solving Guide and User Manual Tutorials (see the demo summary).

Scenario 2 Background (radar example)

The radar example deals with a mountainous terrain area and the drain lines and break lines for this area. The set consists of four files: a 200m spacing Digital Elevation Model (DEM) generated by Vexcel, a resampled "west look" 48 m pixel radar image, and two vector files describing drain and break lines, in CP3 line format. The image covers an area that is common to all products. This area is specified in UTM coordinates (zone 18). The DEM covers a slightly larger area and the vector files even larger. All files were to be imported using their full coverage, leaving it up to the GIS to determine overlap; this feature works fine in GRASS.

For this data set we investigate the following:

- . simple combination maps of the data products together
- . perspective views of the DEM, possibly with imagery or layers draped
- . editing of the drain and cell lines with examination and comparison of results
(ie. connecting drain lines to form a complete drainage network)
- . creation of features using the image and comparing with existing vector files
- . creation of derived products such as slope and aspect, watershed or helicopter landing sites

These steps can all be performed within GRASS, but first the input files, as produced by the Vexcel "mapping" group, must be transformed into GRASS-compatible formats for import. The following section discusses this process.

The input files are the following:

/usr/radian/west8.pic	(radar image)
/usr/radian/dma/final9	(DEM)



EOS Workstation G.R.A.S.S. Investigation Report

/usr/radian/dma/input/west_all.bre *(break lines)*

/usr/radian/dma/input/west_all.dra *(drain lines)*

Scenario 2 External Preprocessing

Two kinds of conversions must be made for this data set: raster and vector.

Raster Input Format

The image and DEM inputs are essentially rasters - they have a rectangular grid of cells with a value at each cell. The image, "west8.pic", is a binary file, 1 byte per element consisting of 1163 rows each 1232 pixels wide, in column order (size 1232x1163=1432816 bytes). The DEM is formatted as an ASCII file one value per line in column order:

1325

1355

2212

.

.

.

Vector Input Format

The drain and break lines are in vector (or point) form. Each "line feature" is a set of file lines delineated as a header with a sequence number and then a number of x,y point lines following. The sequence numbers have no meaning.

GRASS ASCII Raster Input Format

This format consists of N ascii values per line; N can be any number but must be consistent throughout the file. We have chosen to put "1" value per line. All values are in column-order. A header must precede the file (described in the preprocessing section).

GRASS ASCII Vector Input Format (digit file)

This format is described in the GRASS document "Digit File Format" and consists of two files: one is the points file defining each feature line as a sequence of points; the second is an attribute file consisting of 1 ASCII line per line feature in the line feature file. This line describes the type of feature (Area or Line but we use only Line) and the placement of a feature label. In our case all label locations must be either a point on the feature line or closer to that line than any other. The program to produce this file takes each line and places the label in the center of the line using a simple average (see the preprocessing section).

Raster Preprocessing

Since GRASS requires ASCII input files, the binary image file must first be converted. A Vexcel program, "toddecim" can be run to convert a binary file of bytes to its equivalent in decimal ASCII, one value per line. The program uses standard I/O. The following command was used for processing the image:

toddecim < /usr/radian/west8.pic > Jwest8.dec.



EOS Workstation G.R.A.S.S. Investigation Report

Now the image file and DEM file can be edited to contain GRASS headers and then imported into the GRASS environment. Each header must be of the following form:

```
north:
south:
east:
west:
rows:
cols:
<blank line>
```

The two files are described as:

image: north: 9005190.0
south: 8949330.0
east: 394074.0
west: 334932.0
nrows: 1163
ncols: 1232

DEM: north: 9005200.0
south: 8949200.0
east: 394200.0
west: 334800.0
rows: 281
cols: 298

At this point these files are ready for import to GRASS.

Vector Preprocessing

Vectors must first go through another Vexcel program called *tovect* which translates the mapping output format for drain and break lines into the GRASS import format (GRASS ascii vector digit files). There are different versions of "tovect" for drain and break lines since the formats are slightly different. Basically the program counts the number of points in a line and places a header line before each of these line sets indicating that the feature is a line (L) and the number of points in the line:

```
L    2
3.44  4.66
2.1   1.003
L    1
1.0   2.0
.
.
```

Additionally, an attribute file is required which describes each feature in the feature file. One line

EOS Workstation G.R.A.S.S. Investigation Report

per feature is required consisting of a feature type (L) and a location indicating where to place a category label for the feature (UTM coord.s); finally a category number is required to describe the feature:

L 4.00 3.2 10

The label location **must** be a point on the feature line or else closer to it than any other line. The program **tovect** will generate this file along with the feature file. The break and drain lines can then be imported.

NOTE: one further step that may be required (in our case it was) is to sort the points and determine the minimum and maximum geographic values in the sets. The GRASS **digit** program will do this automatically but it is a good idea to know the coverage beforehand. Also, this allows the editing of bad features (ie. single point lines in our case).

Scenario 2 Grass Importing and Preprocessing

There are two import methods, based on type - vector or raster. We discuss the raster case first.

Raster Import

The GRASS program **Mimportcell** is run to input the ASCII cell files. This program requires a filename as a parameter (non-interactive command with a filename and map layer title). The image file is imported as the file **west8**. Cell files have a number of support files associated with them for color tables, histogram, location and category. The DEM is imported similarly and is given the cell map name **final_DEM**. At this point the files can be manipulated by any of the appropriate GRASS programs.

Vector Import

Vector import uses the program **build.vect** to create vectors. Then the program **support.vect** must be run to extract topological information which is required by the **digit** program. **digit** is run to assign labels to the vectors. At this point the vectors are in the database and can be converted to cell files for further manipulation.

Scenario 2 GRASS Manipulation and Analysis

GIS Application Design

The initial objective of this scenario was to have all the data sets (vector and raster) available for use within the GRASS system. A number of different "sub-scenarios" were possible. This section describes various intermediate products required to produce end-results or products that can be visualized (next section). Ultimately we are seeking to answer one or more questions about an area of interest. In our case we have a mountainous area with break and drain line and we are interested in the following questions:

1. When viewing the image with its vectors, either in 2D or as a 3D perspective view, are the vectors consistent with the physical scene depicted? More specifically, do break lines seem like they are correctly placed? Do drain lines all go down hill?

EOS Workstation G.R.A.S.S. Investigation Report

2. Can we draw the vectors ourselves over the imagery without the use of the vectors and come up with features that are consistent with the vectors to a reasonable tolerance? Can we view the results of this process with the GIS?

3. Are there any derived maps that can be generated from our data that provides information not available outside of the GIS?

4. Using the GIS can we derive additional questions concerning the fidelity of the data that we are manipulating? Can we subsequently answer these questions using the GIS?

Processing Steps

In order to answer the above questions we must first decide what GRASS tools can be used and what processing is required to convert our data to the correct formats?

The first question can be answered easily. First we can do a simple display of the radar image and then overlay the line features. In order to display the image we can use the display function *Dcell* (see the following section). The vectors can be overlaid with *Dvect*. Note that the best results for the current image were obtained by using the function *reclass* to reclassify the dynamic range of the file from 0...255 to 1-230. This allows correct sharing of the colortable with other GRASS menu functions. A histogram stretch was also used to bring out details (color table selection #5).

Alternatively, we can use the DEM to form a 3D perspective model with a category map optionally overlaid. This process works reasonably well when viewed from a distance but due to the 200m resolution of the DEM the features appear jagged up close, especially with a thematic map draped over it. No processing of the DEM is required for this. Points of view can be chosen interactively.

The *watershed*, *slope.aspect*, *combine*, *distance* and *Gmapcalc* functions were used to combine and manipulate the various products in forming various representations of the data. These in turn led to further insights as discussed in the conclusions section.

Scenario 2 Results Visualization and Reporting

Viewing the results of the the various processing steps can be performed using the various display tools provided by GRASS. These include the interactive programs *display*, *d.colors*, *d.window*, *d.his*, and *d.3d* for zooming, display, color table manipulation, IHS mapping and perspective view generation. Reports can be generated using the functions *cell.stats*, *layer.info*, *describe* and *report*.

Scenario 2 Conclusions and Follow up Processing

A natural feedback loop exists from the GIS processing back to the DEM production phase. Problems with the DEM or vector files can be identified using GRASS and corrected back in the production environment. These corrections then can be rechecked by the GIS in forming an important verification cycle.

In particular, we have indentified the following additional issues:

1.) Can we compute the mean-slope and standard deviations under drain line areas using, for



EOS Workstation G.R.A.S.S. Investigation Report

instance the *neighbors* program to determine whether the drains lines are believable or correct?

- 2.) Can we use the *watershed* program on steep areas to verify that drainages in fact stop at break lines (ie. ensure that water does not flow over mountain peaks)?
- 3.) By using the *watershed* program with various parameters, are we able to match up the coded stream network features with drain lines? Is this process useful in determining areas of poor DEM quality (noisy DEM, non-matching features)?
- 4.) If we iteratively choose watershed outlet points and apply the watershed model, are we able to trace the same drain/river lines as the mapping features? Do we identify additional rivers or invalid drain lines during this process?
- 5.) Can we use proximity analysis to perform a litmus test of sorts to tell us whether the drain lines provided by the mappers falls within the bounds of the watershed model (ie. an all -yellow boundary becomes all-blue if the coverage is complete)?

The answer to all of the above appears to be yes; - therefore the GIS seems to be a useful and desirable option for supplementing at least the mapping activities at Vexcel. Implementation of these operations and integration into production would take some work but would be feasible with the GRASS GIS and hopefully others as well. Similar arguments can probably be made for other company activities, for example, contour generation, image rectification, map registration, etc.

Scenario 3 Background (AVIRIS bands)

This example is unimplemented but it is expected that the EOSW effort will include a revisit of GIS capabilities near the end of the Phase II development and at this time AVIRIS data will be examined. Many possibilities exist for this data including geologic map registration and cluster analysis.

References

- Ehlers, M., Edwards, G., Bedard, Y., 1989. Integration of Remote Sensing with GIS: A Necessary Evolution. *PERS* Vol. 55, No. 11, pp. 1619-1627.
- Kasturi, R., Fernandez, R., Amlani, M.L., Feng, W., 1989. Map Data Processing in Geographic Information Systems. *IEEE Computer*, Vol. 22.
- Knapp, E., 1978, Landsat and ancillary data inputs to an automated geographical information system. Report No. CSC/tr78.6019, Computer Science Corp., Silver Springs, Maryland.
- Smith, T. R., Menon, S., Star, J., and Estes J., 1987. Requirements and principles for the implementation and construction of large-scale geographic information systems. *Int. J. of GIS*, Vol.1, # 1, pp. 13-31.

GRASS 3.0 PROGRAMS

ABSTRACT

This document lists the currently running programs associated with the Geographical Resources Analysis Support System (GRASS) developed at the U.S. Army Corps of Engineers Construction Engineering Research Laboratory (USA-CERL), Champaign, Illinois.

1. GRASS

GRASS programs come in both interactive and non-interactive forms. Those who have worked with GRASS over the past years will recognize the friendly interactive approach. GRASS 3.0 is the first release to offer a full complement of non-interactive map analysis and display functions. These programs are available to those wishing to write map analysis models and for others desiring to provide new interactive front-ends to GRASS utilities.

The following sections are divided as follows:

- 1 - Map development programs
- 2 - Interactive analysis/display programs
- 3 - Non-interactive analysis programs
- 4 - Non-interactive display programs
- 5 - Image analysis programs

2. Map Development Programs

This set of GRASS tools is designed for the input, manipulation, and adjustment of data from sources outside of GRASS.

2.1. Digitizing

These tools allow for the addition of paper (and other hardcopy geographical information) into GRASS data bases.

digit - Map development digitizing program. Interactive digitizing program for converting hard-copy maps into digit files which can be used to create dlg files.

a.b.dlg -

Convert ascii dlg-3 file to binary dlg-3 file.

b.a.dlg -

Convert binary dlg-3 file to ascii dlg-3 file.

a.b.vect -

Convert ascii vector (digit) files to binary vector (digit) files in current directory.

b.a.vect -

Convert binary vector (digit) files to ascii vector (digit) files in current directory

build.vect -

Converts binary digit file into GRASS Vector Format. Creates a dig_plus support file that contains topological information derived from the digit file.

vect.to.cell -

MAPDEV program to convert a vector file to a cell file.

support.vect -

Converts binary vector (digit) file into GRASS Vector Format. Creates a dig_plus support file that contains topological information derived from the vector (digit) file. Will also allow creation of cats file

import.to.vect -

MAPDEV program which performs all necessary processes needed to convert ascii dlg files, binary dlg files, ascii vector (digit) files or binary vector (digit) files into the GRASS Vector Format.

Vpatch -

Patch 2 or more vector files together creating a composite.

Vstat -

Generate statistical information about a digit file.

2.2. Digital Elevation Data

Various tools for manipulating digital elevation models from Defense Mapping Agency (DMA) tapes and USGS Digital Elevation Model (DEM) tapes.

Mdem.examine -

Provides brief description of data contained on a DEM terrain elevation tape.

Mdem.extract -

Extracts (DEM) terrain elevation data that falls into the users current window from 1/2 inch magnetic tape.

MdmaUSGSread -

Extracts DMA data from tapes purchased from the USGS.

Mdshift -

Performs a datum shift. Returns latitude and longitude values based on an output datum from latitude and longitude values based on an input datum.

Mdted.examine -

Provides brief description of data contained on a DMA terrain elevation tape.

Mdted.extract -

Extracts data from DMA terrain elevation tape.

Mgc2ll -

Conversion of geocentric to geographic coordinates. Geocentric coordinates are measured in meters, in three dimensions, from the center of the earth. Geographic coordinates are in latitude and longitude.

Mimport.ll -

Creates a cell file from data in latitude,longitude coordinates.

Mimportcell -

Converts an ascii text file into a cellfile.

Mll2gc -

Conversion of geographic to geocentric coordinates. Geographic coordinates are in latitude and longitude. Geocentric coordinates, in meters, are measured in the three dimensions, x, y, and z, from the center of the earth.

Mll2u -

Conversion of geographic to UTM coordinates. Geographic coordinates are in latitude and longitude. UTM coordinates are in northings and eastings.

Mrot90 -

Will rotate matrix files 90 degrees. Useful for orienting data which has "west" at the top of the matrix so that "north" is at the top.

Mu2ll -

Conversion of UTM to geographic coordinates. UTM's are in northings and eastings. Geographic coordinates are in latitude and longitude.

Mwindow.ll -

Displays the user window in lat/long. Gives number of meters per arcsecond at each edge.

3. Interactive Analysis and Display

These GRASS tools represent the original GRASS concept. They are highly interactive analysis and display programs.

basins.fill -

Generates subbasin areas given a coded stream network and a map layer denoting all ridges (including watershed boundary).

cell.stats -

Lists area statistics for map layers. For each category, gives total area in acres, hectares, and sq. miles.

clump -

Groups physically discrete areas into unique categories

coin - Coincident tabulation. Charts the mutual occurrences (coincidence) of all categories from one cell map with all categories of another cell map.

combine -

Map overlay tool which allows combinations of map categories from several maps. Allows combinations using AND, OR, GROUP. Created maps can be overlaid and/or saved with the commands NAME, COVER, and OVER.

compress -

Compresses grid cell files using a run length encoding method. Recommended for keeping disk usage down. Files created in GRASS 3.0 are automatically in a compressed format.

copy - Makes copies of database files. Copies map layers, windows, vector files, etc. Files from any mapset can be copied into the current mapset.

decompress -

Decompresses cell files that have been compressed. Generally not useful except for copying cell files to another GIS.

describe -

Prints a list (or range) of values which occur in a data layer.

display -

Multi-purpose graphics monitor program for displaying map layers. Map editing functions include vector map overlays, labeling, drawing, cell digitizing.

distance -

Proximity analysis. Creates a cellfile based on distances from selected categories.

examine.tape -

Scans a 1/2 inch magnetic tape reporting the number of files and the size of physical blocks.

exit - Exit from GRASS.

gisenv -

Show the current GRASS environment, including the database LOCATION, GISDBASE and MAPSET.

grass.armsted -

Menu driven user interface from GRASS to ARMSED, a simulation model which calculates water and sediment runoff for a given watershed. User must complete 'watershed'

series before running this program.

grass.bnoise -

Interfaces the grass system with a noise simulation model called bnoise. Allows user to choose and/or input information to run the bnoise model, and uses bnoise output to produce a grass map layer depicting noise contours for the given input.

grow -Generates output layer which has areas that are one pixel bigger around than input layer. User may specify output to consist of 0's and 1's (completely binary).

help -The on-line help facility.

layer.info -

Lists detailed information about a chosen map including size, source, creation date, creator, etc.

list - Lists the cell files, vector files, and other database elements available according to the current MAPSET path.

manual -

Provides access to the on-line GRASS REFERENCE manual.

mapmask -

Map generation tool which creates a map containing polygonal areas of interest as designated by the user. The user may designate CIRCLES, REGULAR POLYGONS, or IRREGULAR POLYGONS. The map will reflect generic areas or 'cookie-cutter' areas from another map layer.

mapsets -

Allows user to choose which of the available mapsets shall be searched for desired maps.

mask -

User chooses what available cell file's categories shall be used to identify the current masking layer. Allows removal of the mask too.

monitor -

Allows the user to start, select, list, query the status of, relinquish control of, and stop available graphics monitors, through a series of menus.

neighbors -

An interactive program to enhance or subdue data values by modifying a data value based on its surroundings.

paint -

General purpose access to hardcopy color output. Allows the user to produce a hard copy scaled maps directly from cell layers and vector files, with text overlays.

patch -

Assigns data to "no data" areas in one cell map with data from other cell maps. Useful for updating maps or combining adjacent data layers.

random -

Creates a cell file consisting of random pixels, and an optional sites list. Useful for defining field locations for collecting random samples.

reclass -

Map generation tool which creates maps by recategorizing existing cell maps. The user provides the old to new classification conversion table.

remove -

Removes existing map layers, windows, vector files, etc. Only files in the current mapset can be removed.

rename -

Changes the name of existing map layers, windows, vector files, etc. Only files in the current mapset can be renamed.



report -

Lists area statistics for map layers. Statistics for a map layer correspond to an area of interest defined by WINDOW or AREA_MASK. For each category, total area is given in one or more units of #cells, %cover including/excluding no data cells, acres, hectares, sq. kms and sq. miles.

resample -

Resamples a cell file into the current window

rescale -

Map generation tool which creates maps by rescaling existing cell maps.

sites -Tool for point data. Provides statistical analysis of site occurrence.

slope.aspect -

Generates slope and aspect maps from elevation data. The elevation layer specified by the user must contain true elevation values, not rescaled or categorized data.

support -

Data layer support interface. Allows user to edit the header file, stats file, category file, color table, and history file for data layers in the current mapset.

watershed -

Menu driven program providing steps to complete a watershed analysis, deriving a stream network and a watershed depiction from digital elevation data.

weight -

Map overlay tool which allows combinations of map categories from several maps. The user assigns weights to each category for several maps. Weight creates a new map based on the weights assigned to the categories of each cell.

window -

Window management tool. Allows users to define and request windows.

d.3d -Provides a 3-D display of a map layer on the monitor.

d.colors -

Interactive program for changing category colors of a cellfile.

d.digit -

Digitize lines, areas, and circles on the display monitor, and store this in a cell file.

d.his -Combines data from up to three map layers using Hue, Intensity, and Saturation color components.

d.sites -

Displays point sites on map.

d.window -

Interactive management tool for setting the current window by using a pointing device (mouse) to indicate where on the display device (monitor) the current window is to be drawn.

4. Image Analysis

In past versions of GRASS image analysis was done as a separate subsystem using a completely different data base structure. With GRASS 3.0 image analysis is integrated with the analysis and display capabilities.

i.cluster -

An imagery function that generates spectral signatures for land cover types in an image using a clustering algorithm. The resulting signature file is used as input for i.maxlik.

i.colors -

An imagery function that creates colors for imagery groups

i.composite -

An imagery function that creates a color composite image from three band files specified by the user

i.grey.scale -

An imagery function that assigns a histogram contrast stretch grey scale color table to a map layer

i.group -

An imagery function that creates and edits groups and subgroups of imagery files

i.maxlik -

An imagery function that classifies the pixel spectral reflectances in imagery data based on the spectral signature information generated in *i.cluster*

i.points -

An imagery function that enables the user to mark coordinate system points on an image to be rectified and then input the coordinates of each point for creation of a coordinate transformation matrix. The transformation matrix is needed as input for the GRASS program *i.rectify*.

i.rectify -

An imagery function that rectifies an image by computing a coordinate transformation for each pixel in the image using the transformation coefficient matrix created by the GRASS program *i.points*

i.tape.mss -

An imagery function that extracts Multispectral Scanner Imagery from half inch tape

i.tape.other -

An imagery function that extracts scanned aerial imagery (NHAP, etc.) and SPOT imagery from half inch tape

i.tape.tm -

An imagery function that extracts Thematic Mapper imagery from half inch tape

i.target -

An imagery function that establishes a GRASS-GRID target LOCATION for an imagery group

5. Non-interactive Screen Graphics

These utilities provide the modeler and GRASS interface developer with assorted graphics capabilities.

D3d -Provides a 3-D display of a map layer on the monitor.

D3d.view -

A Bourne shell script presenting several 3-d views of a user-selected cell file on the monitor.

Dcell -Display grid cell maps in current graphics window

Dchoose -

Choose (i.e., select) an available graphics window in which output will be displayed.

Dclear.screen -

Clears the entire screen, removing all traces of existing windows

Dcolormode -

Switches between fixed and floating color lookup tables.

Dcolortable -

Allows the user to choose a color look-up table to be associated with the display of graphics



- Derase -
Erase the contents of the current graphics window with the specified color.
- Dfont -
Changes the font type to that specified by the user
- Dgraph -
Draw simple graphics in current graphics window.
- Dgrass.logo -
A macro using Dgraph commands to generate and display the GRASS logo
- Dgrid -
Draws a grid of the specified size and color in the current graphics window.
- Dlabel -
Draws a label of specified color, size, and font in the current graphics window
- Dlegend -
Display legend for a grid cell map in the current graphics window.
- Dlist.mon -
List GRASS graphic output device drivers (monitors).
- Dmapgraph -
Draw simple graphics on map in current graphics window.
- Dmeasure -
Measure the lengths and areas of lines and polygons drawn on the monitor
- Dmenu -
Draws a menu of specified size and color on the monitor
- Dnew -
Create a new graphics window on the monitor.
- Doverlay -
Overlay grid cell maps in current graphics window. Only non-zero data values are displayed.
- Dpaint.labels -
Display label files generated by the paint label utility.
- Dpoints -
Displays a list of point coordinates on a map.
- Drelease.mon -
Forces control of the named graphics output device (monitor) to be relinquished, whether or not it is locked by another user.
- Dremove -
Removes a graphics window (other than the current window) from the monitor.
- Dsavescreen -
Saves the current image on the graphics monitor in a file for later painting by Pscreen.
Only available to those with MASSCOMP systems.
- Dscale -
Places a scale of named color and size on a map in current graphics window.
- Dscreen -
Easy way to erase the entire screen, create a new window covering the entire screen, and choose this full screen window as the current window.
- Dselect.mon -
Selects a graphics device (monitor) for output, initializes the screen, and locks the device for use by the user

Dshow.fonts -

Shows the user fonts that can be selected using Dfonts.

Dslide.show -

Displays maps in the user's mapset on the monitor.

Dstart.mon -

Loads the specified graphics device driver program (also called a monitor) into memory.

Dstatus -

Prints to standard output information about the windows on the monitor.

Dstatus.mon -

Lists the status of the programs which control graphic output devices. These programs are called device drivers or monitors. The status may be In Use (locked by another user), Idle (in memory but not locked), or Not Loaded (not resident in memory).

Dstop.mon -

Can terminate the named graphics device controller program (also called a monitor) even when it is locked by another user when run with the appropriate option.

Dtext -

Draws text of the named size and color in the current graphics window.

Dtitle -

Creates output which is suitable for input to Dtext, to create a title for a grid cell map of the named size and color.

Dvect -

Display vector (digit) maps in current graphics window.

Dvect.dlg -

Display dlg-3 maps in current graphics window.

Dwhat -

Interactive program printing out the category numbers and labels associated with specified maps at a grid cell location pointed to by the user.

Dwhat.once -

Runs the Dwhat command once, showing the user what category number and label are located on a map at the spot pointed to by the user (useful for programs which require one input from the user).

Dwhere -

Interactive program printing out the map coordinates associated with map locations pointed to by the user.

Dwhich -

Identifies which window contains a specified screen coordinate.

Dwhich.mon -

Determines which graphics output device is currently selected for output, and states this.

6. Non-interactive Analysis

These utilities provide the modeler and GRASS interface developer with assorted analysis capabilities. Most of these have counterparts in the interactive programs listed above. Infact, many of those programs use the corresponding G function to do the actual work.

Gask -

Prompts the user for database files. Used by shell scripts that need to prompt the user.

Gclump -

Command line interface for grouping physically discrete areas into unique categories

Gcopy -

Makes copies of database files. Copies map layers, windows, vector files, etc. Files from any mapset can be copied into the current mapset.

Gcost -

Evaluates the cost of traveling from one or more starting points to each grid cell in the window by traveling over a cost surface. Cost surfaces can be topographic elevation, roads, soils etc.

Gcross -

Produces a cross product of multiple data layers. The resulting data layer contains a unique category for every combination of values in the input data layers.

Gdescribe -

Prints a list (or range) of values which occur in a data layer.

Gdistance -

Command line interface for proximity analysis. Creates a cellfile based on distances from selected categories.

Gdrain -

Finds the lowest cost path over a surface from starting points. The surface can be elevation (in which case the path will be that which a raindrop would follow to the low point) or a cost surface (resulting in the least cost path).

Gdumpcell -

Gdumpcell converts a cell file into a ascii text file. Useful for file exportation to other computer systems or quickly viewing an ascii version of a cell file.

Gfindfile -

Gfindfile is designed for shell scripts that need to search for cell files, vector files, site list files, window files, and imagery group files in the data base.

Ggrow -

Adds one pixel to the border of areas in a map layer.

Ginfer -

Map "expert-system" tool which allows combinations of map categories from several maps.

Glayer.info -

Glayer.info retrieves layer information for cell files, and results in the production of a table listing various information about a user chosen map layer.

Glist -Lists the cell files, vector files, and other database elements available according to the current MAPSET path.

Glos -This line-of-site tool generates a map of those cells which are visible from a user specified location.

Gmapcalc -

Full arithmetic expression map calculator. Used to produce new map layers from arithmetic combinations of existing map layers.

Gmapsets -

Gmapsets sets the current mapset search path to include whatever mapsets are named on the command line. These mapsets can then be accessed by the user.

Gmfilter -

Filters the input to produce output according to the matrix filter designed by the user.

Gpat.place -

Gpat.place allows a user to place a pattern of his choice at a particular location and in a desired direction.

Gpatch -

Assigns data to "no data" areas in one cell map with data from other cell maps. Useful for updating maps or combining adjacent data layers.

Grandom -

Creates a cell file consisting of random pixels, and an optional sites list. Useful for defining field locations for collecting random samples.

Greclass -

Map generation tool which creates maps by recategorizing existing cell maps. The user provides the old to new classification rules.

Gremove -

Removes existing map layers, windows, vector files, etc. Only files in the current mapset can be removed.

Grename -

Changes the name of existing map layers, windows, vector files, etc. Only files in the current mapset can be renamed.

Greport -

Greport allows the user to set up a series of report parameters which can then be applied to one or more map layers, creating a report providing the requested statistics for each grid cell map layer.

Gresample -

Command line interface for Resampling a cellfile into the current window

Grescale -

Grescale creates a new data layer that is a rescaling of an existing data layer.

Gsites -

Gsites lists locations of points that are contained in the specified GRASS sites file. Point locations are listed as pairs of grid coordinates in the form of Eastings and Northings. Gsites output can be redirected directly into the Dpoints program.

Gslope.aspect -

Generates slope and aspect maps from elevation data. The elevation layer specified by the user must contain true elevation values, not rescaled or categorized data.

Gstats -

Prints area statistics for user specified grid cell data layers. Applied to a single layer, the area for each category which occurs in the layer is printed. Applied to multiple layers, an area coincidence table across all the layers is printed.

Gsurface -

Gsurface allows a user to generate a smooth interpolated surface from an input of irregularly spaced values with intervening areas of no data (i.e. zeros).

Gtraj -

Gtraj allows a user to place a piece of artillery at a specified location and give the charge the necessary characteristics required to generate a cell map of the resultant trajectory.

Gweight -

Gweight is a weighted overlay analysis tool, and is the non-interactive version of weight(1). Like weight, Gweight allows the user to assign numerical weights to each of the categories into which grid cell map layers are classified.

Gwindow -

Window management program. Non-interactive interface for setting and printing the current window.

Pmap -

Command language interface to hardcopy color output

Pscreen -

Paint screen image file saved by Dsavescreen (if your system has Dsavescreen)

Pselect -

Selects hardcopy output device.

Vexcel Corporation

EOS Workstation

**DEMO
of
USACERL's**

Geographical Resources Analysis Support System

(GRASS)

GRASS 3.1 Demo

* - to display the following the command "setenv GRASS_COLOR256" must be performed.

Scenario I - GRASS Sample database: spearfish, South Dakota

This example is meant to show GRASS basics, in particular for an area with many thematic maps. This data set has cell map layers for forestry, elevation, ownership, roads, soils, hydrography, vegetation and many more. There are also vector files for farm fields, railroads, streams and transportation. The first examples will deal with simple cell map displays and vector overlays and are meant to indicate the types of situations typical of GIS applications. The second example set will deal with a complex multi-layer set.

Cell and Vector Maps

The following are of interest and can be displayed with *Dcell* and *Dvect*:

cells:	mss.image	vectors:	roads
	geology		streams.spear
	landuse		railroads

Multi-layer Example

This example demonstrates the use of a more complex application. It is example 6 of the GRASS Problem Solving Manual, "Predicting Forest Fire Potential". This example predicts the wildfire potential of all forested areas for the spearfish database. The results show 3 categories of fire potential: *low, medium and high*. This map was created using the GRASS functions **Gmapcalc**, **distance**, **weight** and **reclass**. Overlaid for reference on this map are the roads, stream and railroads for this area. Additional vectors can also be overlaid. The variables used in this example are type of forest, presence of high crown density, presence of dead wood and presence of steep slopes in high density forest. See the example in the

problem solving details of how to form this map.

Scenario II - Vexcel Radar Mapping

This data set consists of original and derived products. The original sets include a radar image (**west8**), a DEM (**final9_DEM**) and two line feature (vector) files for break (**west_all.dig**) and drain (**west_all_d.dig**) lines. Again simple examples are shown followed by a more complex application with the intention of demonstrating useful GIS functions:

Simple Examples

1. The most basic example to use is simply to view the imagery with the vector files overlaid. The low contrast of the original image is improved by reclassifying the data into a smaller number of values to avoid color-table conflicts and then doing a histogram stretch based on the values. This new image with 230 grey values can be displayed with the break and drain lines overlaid. This can be done with Dcell and Dvect or with the interactive program **display**.
2. Another display is the perspective view of the DEM with/without categories. This can be done using the GRASS utility **display**.
3. A third simple example utilizes the **digit** function of GRASS, a general purpose digitizer. This function allows editing of vectors with an image backdrop. Zooming, labelling, digitizing and many other functions are supported by this program.

These examples primarily demonstrate the system's visualization capabilities for different types of data.

Derived Data sets

Variety exists for deriving data sets from the original 4 products: aspect and slope models can be produced from the DEM, thematic maps can be created using the **watershed** tool and the DEM, arithmetic and logical operations can be performed on the imagery and on cell files created from vector files. This demo presents the following derived data sets:

1. Slope and aspect models with drain and break lines overlaid
2. Watershed products with drain and break lines overlaid
3. User-defined river features with a distance analysis combined with a watershed area and coded stream network - all overlaid with the drain and break points.

The last example is in fact part of the complex example at the end of this document. It is described there. The following discusses the slope example:

Slope and aspect layers -

The slope.aspect program takes a DEM as input and produces a slope and aspect layer. The slope layer contains a 1 for 0 degrees slope, a 2 for 1 degree, etc. A color table must then be generated for this file so it can be displayed as a grey scale file. This file, when viewed with the break and drain line files gives a good approximation of terrain variation with respect to the acquired features. The aspect file gives values such as:

- 1 - east facing
- 2 - 15 degrees north of east
- 3 - 30 degrees north of east
- 4 - northeast facing
- .
- .
- .
- 25 - no aspect (flat)

Further manipulations of the slope and aspect files are also possible.

Complex Application

The final demo set combines a number of elements into a final product. To begin with we use the DEM to run the watershed program. This program produces the following products (using only an elevation model):



- . filtered DEM
- . drainage accumulation map layer
- . adjusted drainage direction map
- . stream network and coded stream network
- . watershed basins

We used 7m deep pits and an outlet point obtained from looking at the radar image and using Dwhere (E370474, N8981305).

Once the watershed modelling is completed we can, using Gmapcalc, combine the radar image with the watershed and stream network files, assigning the image to a grey scale and the watershed and stream files to light blue and white.

In parallel we use the *digit* program to trace the path of two river lines on the radar image. We then label these lines and save them in a vector file. This file is converted to a cell file and applied to a proximity analysis with a value of 240 meters. This produces a cell map of the river line with a 240m border around it. We combine this layer with the previous layer and assign it a yellow color.

For the last step we overlay the original drain and break lines in blue and red respectively, as well as the new river file (before it was converted) in green. The result is a layer which shows on the original imagery the watershed and stream network as well as the original drain and break lines and new drain lines and border. From this we can verify two things:

1. Do all original drain lines fall within the 240m border derived from the user-defined river area? (The answer is yes!)
2. Are there places along the user-defined river that are not covered by the original drain lines? (The answer is yes - these are areas that did not provide stereo coverage (were in shadow) and therefore no stereoplotter acquisition was done.)?

These examples demonstrate the usefulness of a GIS to some of the conventional activities performed at Vexcel - in particular data sets similar to those required for the EOS Workstation. Additional issues are addressed in the GRASS Investigation report.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
84

MAP LAYER CATEGORY REPORT

ayer: [geology] in mapset [PERMANENT]
 Location: spearfish
 Title: Geology
 Mask: <MASK> in mapset <PERMANENT>

Date: Thu Sep 13 15:31:22

Window: north: 4928000.00 east: 609000.00
 south: 4914000.00 west: 590000.00
 res: 50.00 res: 50.00

cat#	Name	# cells	% w/ cat 0	% no cat 0
0	no data	45132	42.42	0.00
1	metamorphic	3060	2.88	4.99
2	transition	52	0.05	0.08
3	igneous	11632	10.93	18.99
4	sandstone	19612	18.43	32.01
5	limestone	16240	15.26	26.51
6	shale	2700	2.54	4.41
8	claysand	4648	4.37	7.59
9	sand	3324	3.12	5.43
totals		106400	100.00	100.00

cat#	Name	acres	hectares	sq kms
0	no data	27880.29	11283.00	112.83
1	metamorphic	1890.31	765.00	7.65
2	transition	32.12	13.00	0.13
3	igneous	7185.67	2908.00	29.08
4	sandstone	12115.31	4903.00	49.03
5	limestone	10032.26	4060.00	40.60
6	shale	1667.93	675.00	6.75
8	claysand	2871.30	1162.00	11.62
9	sand	2053.40	831.00	8.31
totals		65728.60	26600.00	266.00

cat#	Name	sq miles
0	no data	43.56
1	metamorphic	2.95
2	transition	0.05
3	igneous	11.23
4	sandstone	18.93
5	limestone	15.68
6	shale	2.61
8	claysand	4.49
9	sand	3.21
totals		102.70

MAP LAYER CATEGORY REPORT

Layer: [vegcover] in mapset [PERMANENT]
 Location: spearfish
 Title: Vegetation Cover
 Mask: none

Date: Tue Sep 18 17:33:27

Window: north: 4928000.00 east: 609000.00
 south: 4914000.00 west: 590000.00
 res: 50.00 res: 50.00

cat#	Name	# cells	% w/ cat 0	% no cat 0
1	irrigated agriculture	10216	9.60	9.60
2	rangeland	32900	30.92	30.92
3	coniferous forest	40348	37.92	37.92
4	deciduous forest	8696	8.17	8.17
5	mixed forest	10976	10.32	10.32
6	disturbed	3264	3.07	3.07
totals		106400	100.00	100.00

cat#	Name	acres	hectares	sq kms
1	irrigated agriculture	6310.93	2554.00	25.54
2	rangeland	20323.97	8225.00	82.25
3	coniferous forest	24924.98	10087.00	100.87
4	deciduous forest	5371.95	2174.00	21.74
5	mixed forest	6780.42	2744.00	27.44
6	disturbed	2016.34	816.00	8.16
totals		65728.60	26600.00	266.00

cat#	Name	sq miles
1	irrigated agriculture	9.86
2	rangeland	31.76
3	coniferous forest	38.95
4	deciduous forest	8.39
5	mixed forest	10.59
6	disturbed	3.15
totals		102.70



MAP LAYER CATEGORY REPORT

Layer: [soils] in mapset [PERMANENT] Date: Tue Sep 18 17:36:19
 Location: spearfish
 Title: Soils
 Mask: none
 Window: north: 4928000.00 east: 609000.00
 south: 4914000.00 west: 590000.00
 res: 50.00 res: 50.00

cat#	Name	# cells	% w/ cat 0	% no cat 0
1	Alice fine sandy loam, AaB	32	0.03	0.03
2	Boneek silt loam, BcA	184	0.17	0.17
3	Boneek silt loam, BcB	680	0.64	0.64
4	Butche stoney loam, BeE	140	0.13	0.13
5	Butche-Rock outcrop, BhE	2548	2.39	2.39
6	Butche-Satanta loams, BkD	884	0.83	0.83
7	Canyon-Bridget complex, CaD	500	0.47	0.47
8	Canyon-Bridget complex, CaE	1272	1.20	1.20
9	Citadel assoc., hilly, CBE	1308	1.23	1.23
10	Dumps, mine, Cc	12492	11.74	11.74
11	Enning-Minnequa silty clay loam, EaD	228	0.21	0.21
12	Glenberg Variant fine sandy loam, GaD	168	0.16	0.16
13	Grizzly-Virkula assoc., GBE	16616	15.62	15.62
14	Gummit-Rock outcrop, GcD	80	0.08	0.08
15	Gummit-Rock outcrop, GdE	64	0.06	0.06
16	Gypnevee-Rekop loams, GeD	1432	1.35	1.35
17	Higgins silt loam, Ha	28	0.03	0.03
18	Hisega-Rock outcrop, HBF	124	0.12	0.12
19	Hisle silt loam, HcA	272	0.26	0.26
20	Kyle clay, KaA	2984	2.80	2.80
21	Lakoa silt loam, LaE	36	0.03	0.03
22	Maitland loam, MaC	3912	3.68	3.68
23	Maitland loam, MaD	416	0.39	0.39
24	Midway-Razor silty clay loam, McD	1152	1.08	1.08
25	Nevee silt loam, NaB	1452	1.36	1.36
26	Nevee silt loam, NaC	2860	2.69	2.69
27	Nevee-Spearfish-Rock outcrop, NbD	516	0.48	0.48
28	Nihill Gravelly Loam, NcD	64	0.06	0.06
29	Nunn clay loam, NdA	328	0.31	0.31
30	Nunn clay loam, NdB	180	0.17	0.17
31	Nunn clay loam, NdC	6264	5.89	5.89
32	Paunsaugunt-Rock outcrop, PbE	76	0.07	0.07
33	Pierre Clay, PcB	244	0.23	0.23
34	Pierre Clay, PcD	120	0.11	0.11
35	Pits, quarry, Pe	412	0.39	0.39
36	Rock outcrop-Pactola assoc., RBF	3696	3.47	3.47
37	Rock outcrop-Vanocker assoc., RCF	204	0.19	0.19
38	Satanta loam, SaA	28	0.03	0.03
39	Satanta loam, SaB	448	0.42	0.42
40	Savo silt loam, SbA	16	0.02	0.02
41	Savo silt loam, SbB	924	0.87	0.87

100

cat#	Name	# cells	% w/ cat 0	% no cat 0
42	Stetter Variant silty clay loam, Sd	1552	1.46	1.46
43	St. Onge loam, ShA	380	0.36	0.36
44	Swint silt loam, Sk	1752	1.65	1.65
45	Tilford silt loam, TaA	376	0.35	0.35
47	Tilford silt loam, TaC	3840	3.61	3.61
48	Vale silt loam, VaA	5312	4.99	4.99
49	Vale silt loam, VaB	1164	1.09	1.09
50	Vale silt loam, Vac	17832	16.76	16.76
51	Vanocker-Citadel assoc., VBF	6396	6.01	6.01
52	Weber loam, WaA	684	0.64	0.64
53	Winetti cobbly loam, Wb	1708	1.61	1.61
54	Water	20	0.02	0.02
totals		106400	100.00	100.00

cat#	Name	acres	hectares	sq kms
1	Alice fine sandy loam, AaB	19.77	8.00	0.08
2	Boneek silt loam, BcA	113.67	46.00	0.46
3	Boneek silt loam, BcB	420.07	170.00	1.70
4	Butche stoney loam, BeE	86.49	35.00	0.35
5	Butche-Rock outcrop, BhE	1574.03	637.00	6.37
6	Butche-Satanta loams, BkD	546.09	221.00	2.21
7	Canyon-Bridget complex, CaD	308.88	125.00	1.25
8	Canyon-Bridget complex, CaE	785.78	318.00	3.18
9	Citadel assoc., hilly, CBE	808.02	327.00	3.27
10	Dumps, mine, Cc	7716.93	3123.00	31.23
11	Enning-Minnequa silty clay loam, EaD	140.85	57.00	0.57
12	Glenberg Variant fine sandy loam, GaD	103.78	42.00	0.42
13	Grizzly-Virkula assoc., GBE	10264.53	4154.00	41.54
14	Gummit-Rock outcrop, GcD	49.42	20.00	0.20
15	Gummit-Rock outcrop, GdE	39.54	16.00	0.16
16	Gypnevee-Rekop loams, GeD	884.62	358.00	3.58
17	Higgins silt loam, Ha	17.30	7.00	0.07
18	Hisege-Rock outcrop, HBF	76.60	31.00	0.31
19	Hisle silt loam, HcA	168.03	68.00	0.68
20	Kyle clay, KaA	1843.37	746.00	7.46
21	Lakoa silt loam, LaE	22.24	9.00	0.09
22	Maitland loam, MaC	2416.64	978.00	9.78
23	Maitland loam, MaD	256.98	104.00	1.04
24	Midway-Razor silty clay loam, McD	711.65	288.00	2.88
25	Nevee silt loam, NaB	896.97	363.00	3.63
26	Nevee silt loam, NaC	1766.77	715.00	7.15
27	Nevee-Spearfish-Rock outcrop, NbD	318.76	129.00	1.29
28	Nihill Gravelly Loam, NcD	39.54	16.00	0.16
29	Nunn clay loam, NdA	202.62	82.00	0.82
30	Nunn clay loam, NdB	111.19	45.00	0.45
31	Nunn clay loam, NdC	3869.59	1566.00	15.66
32	Paunsaugunt-Rock outcrop, PbE	46.95	19.00	0.19
33	Pierre Clay, PcB	150.73	61.00	0.61
34	Pierre Clay, PcD	74.13	30.00	0.30
35	Pits, quarry, Pe	254.51	103.00	1.03
36	Rock outcrop-Pactola assoc., RBF	2283.20	924.00	9.24
37	Rock outcrop-Vanocker assoc., RCF	126.02	51.00	0.51

cat#	Name	acres	hectares	sq kms
38	Satanta loam, SaA	17.30	7.00	0.07
39	Satanta loam, SaB	276.75	112.00	1.12
40	Savo silt loam, SbA	9.88	4.00	0.04
41	Savo silt loam, SbB	570.80	231.00	2.31
42	Stetter Variant silty clay loam, Sd	958.75	388.00	3.88
43	St. Onge loam, ShA	234.74	95.00	0.95
44	Swint silt loam, Sk	1082.30	438.00	4.38
45	Tilford silt loam, TaA	232.27	94.00	0.94
47	Tilford silt loam, TaC	2372.16	960.00	9.60
48	Vale silt loam, VaA	3281.49	1328.00	13.28
49	Vale silt loam, VaB	719.06	291.00	2.91
50	Vale silt loam, VaC	11015.72	4458.00	44.58
51	Vanocker-Citadel assoc., VBF	3951.13	1599.00	15.99
52	Weber loam, WaA	422.54	171.00	1.71
53	Winetti cobbly loam, Wb	1055.12	427.00	4.27
54	Water	12.35	5.00	0.05
totals		65728.60	26600.00	266.00

cat#	Name	sq miles
1	Alice fine sandy loam, AaB	0.03
2	Boneek silt loam, BcA	0.18
3	Boneek silt loam, BcB	0.66
4	Butche stoney loam, BeE	0.14
5	Butche-Rock outcrop, BhE	2.46
6	Butche-Satanta loams, BkD	0.85
7	Canyon-Bridget complex, CaD	0.48
8	Canyon-Bridget complex, CaE	1.23
9	Citadel assoc., hilly, CBE	1.26
10	Dumps, mine, Cc	12.06
11	Enning-Minnequa silty clay loam, EaD	0.22
12	Glenberg Variant fine sandy loam, GaD	0.16
13	Grizzly-Virkula assoc., GBE	16.04
14	Gummit-Rock outcrop, GcD	0.08
15	Gummit-Rock outcrop, GdE	0.06
16	Gypnevee-Rekop loams, GeD	1.38
17	Higgins silt loam, Ha	0.03
18	Hisega-Rock outcrop, HBF	0.12
19	Hisle silt loam, HcA	0.26
20	Kyle clay, KaA	2.88
21	Lakoa silt loam, LaE	0.03
22	Maitland loam, MaC	3.78
23	Maitland loam, MaD	0.40
24	Midway-Razor silty clay loam, McD	1.11
25	Nevee silt loam, NaB	1.40
26	Nevee silt loam, NaC	2.76
27	Nevee-Spearfish-Rock outcrop, NbD	0.50
28	Nihill Gravelly Loam, NcD	0.06
29	Nunn clay loam, NdA	0.32
30	Nunn clay loam, NdB	0.17
31	Nunn clay loam, NdC	6.05
32	Paunsaugunt-Rock outcrop, PbE	0.07
33	Pierre Clay, PcB	0.24

cat#	Name	sq miles
34	Pierre Clay, PcD	0.12
35	Pits, quarry, Pe	0.40
36	Rock outcrop-Pactola assoc., RBF	3.57
37	Rock outcrop-Vanocker assoc., RCF	0.20
38	Satanta loam, SaA	0.03
39	Satanta loam, SaB	0.43
40	Savo silt loam, SbA	0.02
41	Savo silt loam, SbB	0.89
42	Stetter Variant silty clay loam, Sd	1.50
43	St. Onge loam, ShA	0.37
44	Swint silt loam, Sk	1.69
45	Tilford silt loam, TaA	0.36
47	Tilford silt loam, TaC	3.71
48	Vale silt loam, VaA	5.13
49	Vale silt loam, VaB	1.12
50	Vale silt loam, Vac	17.21
51	Vanocker-Citadel assoc., VBF	6.17
52	Weber loam, WaA	0.66
53	Winetti cobbly loam, Wb	1.65
54	Water	0.02
totals		102.70



Layer: [fire.potential] in mapset [PERMANENT]
Title: Fire potential
Mask: none

Window: north: 4928000.00 east: 609000.00
south: 4914000.00 west: 590000.00
res: 50.00 res: 50.00

Category	Acres	Hectares	Sq.mi
0 no data	36998.28	14973.00	57.81
1	7420.41	3003.00	11.59
2	21272.84	8609.00	33.24
3	37.06	15.00	0.06
total	65728.60	26600.00	102.70



MAP LAYER CATEGORY REPORT

Layer: [final9.slope] in mapset [PERMANENT] Date: Tue Sep 18 17:35:03
 Location: Tingo
 Title: slope (in degrees)
 Mask: none

Window: north: 9002000.00 east: 350074.00
 south: 8970012.02 west: 334954.00
 res: 48.03 res: 48.00

cat#	Name	# cells	% w/ cat 0	% no cat 0
0	no data	666	0.32	0.00
1	0 degrees	64	0.03	0.03
2	1 degree	417	0.20	0.20
3	2 degrees	745	0.36	0.36
4	3 degrees	916	0.44	0.44
5	4 degrees	1411	0.67	0.67
6	5 degrees	1589	0.76	0.76
7	6 degrees	2717	1.30	1.30
8	7 degrees	2774	1.32	1.33
9	8 degrees	3054	1.46	1.46
10	9 degrees	3763	1.79	1.80
11	10 degrees	4279	2.04	2.05
12	11 degrees	5107	2.43	2.44
13	12 degrees	5339	2.54	2.55
14	13 degrees	5766	2.75	2.76
15	14 degrees	5908	2.82	2.83
16	15 degrees	6626	3.16	3.17
17	16 degrees	7535	3.59	3.60
18	17 degrees	7217	3.44	3.45
19	18 degrees	7824	3.73	3.74
20	19 degrees	8654	4.13	4.14
21	20 degrees	8806	4.20	4.21
22	21 degrees	8577	4.09	4.10
23	22 degrees	9648	4.60	4.61
24	23 degrees	8856	4.22	4.23
25	24 degrees	8776	4.18	4.20
26	25 degrees	8991	4.29	4.30
27	26 degrees	7986	3.81	3.82
28	27 degrees	7313	3.49	3.50
29	28 degrees	7511	3.58	3.59
30	29 degrees	7594	3.62	3.63
31	30 degrees	6159	2.94	2.95
32	31 degrees	6427	3.06	3.07
33	32 degrees	4760	2.27	2.28
34	33 degrees	5216	2.49	2.49
35	34 degrees	3921	1.87	1.87
36	35 degrees	3049	1.45	1.46
37	36 degrees	2812	1.34	1.34
38	37 degrees	2318	1.10	1.11
39	38 degrees	1817	0.87	0.87
40	39 degrees	1515	0.72	0.72



cat#	Name	# cells	% w/ cat 0	% no cat 0
41	40 degrees	1163	0.55	0.56
42	41 degrees	1049	0.50	0.50
43	42 degrees	720	0.34	0.34
44	43 degrees	590	0.28	0.28
45	44 degrees	388	0.18	0.19
46	45 degrees	272	0.13	0.13
47	46 degrees	333	0.16	0.16
48	47 degrees	136	0.06	0.07
49	48 degrees	200	0.10	0.10
50	49 degrees	160	0.08	0.08
51	50 degrees	16	0.01	0.01
52	51 degrees	100	0.05	0.05
53	52 degrees	96	0.05	0.05
54	53 degrees	64	0.03	0.03
55	54 degrees	16	0.01	0.01
58	57 degrees	48	0.02	0.02
59	58 degrees	16	0.01	0.01
totals		209790	100.00	100.00

cat#	Name	acres	hectares	sq kms
0	no data	379.40	153.54	1.54
1	0 degrees	36.46	14.75	0.15
2	1 degree	237.55	96.14	0.96
3	2 degrees	424.41	171.76	1.72
4	3 degrees	521.82	211.18	2.11
5	4 degrees	803.81	325.30	3.25
6	5 degrees	905.21	366.33	3.66
7	6 degrees	1547.80	626.39	6.26
8	7 degrees	1580.28	639.53	6.40
9	8 degrees	1739.79	704.08	7.04
10	9 degrees	2143.68	867.54	8.68
11	10 degrees	2437.64	986.50	9.86
12	11 degrees	2909.33	1177.39	11.77
13	12 degrees	3041.49	1230.87	12.31
14	13 degrees	3284.74	1329.32	13.29
15	14 degrees	3365.64	1362.05	13.62
16	15 degrees	3774.66	1527.58	15.28
17	16 degrees	4292.50	1737.15	17.37
18	17 degrees	4111.34	1663.84	16.64
19	18 degrees	4457.13	1803.78	18.04
20	19 degrees	4929.96	1995.13	19.95
21	20 degrees	5016.55	2030.17	20.30
22	21 degrees	4886.10	1977.38	19.77
23	22 degrees	5496.22	2224.29	22.24
24	23 degrees	5045.03	2041.70	20.42
25	24 degrees	4999.46	2023.25	20.23
26	25 degrees	5121.94	2072.82	20.73
27	26 degrees	4549.42	1841.12	18.41
28	27 degrees	4166.03	1685.97	16.86
29	28 degrees	4278.82	1731.62	17.32
30	29 degrees	4326.11	1750.75	17.51
31	30 degrees	3508.62	1419.92	14.20

cat#	Name	acres	hectares	sq kms
32	31 degrees	3661.30	1481.71	14.82
33	32 degrees	2711.65	1097.39	10.97
34	33 degrees	2971.42	1202.52	12.03
35	34 degrees	2233.69	903.96	9.04
36	35 degrees	1736.94	702.93	7.03
37	36 degrees	1601.92	648.29	6.48
38	37 degrees	1320.50	534.40	5.34
39	38 degrees	1035.10	418.90	4.19
40	39 degrees	863.06	349.27	3.49
41	40 degrees	662.53	268.12	2.68
42	41 degrees	597.59	241.84	2.42
43	42 degrees	410.17	165.99	1.66
44	43 degrees	336.11	136.02	1.36
45	44 degrees	221.03	89.45	0.89
46	45 degrees	154.95	62.71	0.63
47	46 degrees	189.70	76.77	0.77
48	47 degrees	77.48	31.35	0.31
49	48 degrees	113.93	46.11	0.46
50	49 degrees	91.15	36.89	0.37
51	50 degrees	9.11	3.69	0.04
52	51 degrees	56.97	23.05	0.23
53	52 degrees	54.69	22.13	0.22
54	53 degrees	36.46	14.75	0.15
55	54 degrees	9.11	3.69	0.04
58	57 degrees	27.34	11.07	0.11
59	58 degrees	9.11	3.69	0.04
totals		119511.95	48365.82	483.66

cat#	Name	sq miles
0	no data	0.59
1	0 degrees	0.06
2	1 degree	0.37
3	2 degrees	0.66
4	3 degrees	0.82
5	4 degrees	1.26
6	5 degrees	1.41
7	6 degrees	2.42
8	7 degrees	2.47
9	8 degrees	2.72
10	9 degrees	3.35
11	10 degrees	3.81
12	11 degrees	4.55
13	12 degrees	4.75
14	13 degrees	5.13
15	14 degrees	5.26
16	15 degrees	5.90
17	16 degrees	6.71
18	17 degrees	6.42
19	18 degrees	6.96
20	19 degrees	7.70
21	20 degrees	7.84
22	21 degrees	7.63



cat#	Name	sq miles
23	22 degrees	8.59
24	23 degrees	7.88
25	24 degrees	7.81
26	25 degrees	8.00
27	26 degrees	7.11
28	27 degrees	6.51
29	28 degrees	6.69
30	29 degrees	6.76
31	30 degrees	5.48
32	31 degrees	5.72
33	32 degrees	4.24
34	33 degrees	4.64
35	34 degrees	3.49
36	35 degrees	2.71
37	36 degrees	2.50
38	37 degrees	2.06
39	38 degrees	1.62
40	39 degrees	1.35
41	40 degrees	1.04
42	41 degrees	0.93
43	42 degrees	0.64
44	43 degrees	0.53
45	44 degrees	0.35
46	45 degrees	0.24
47	46 degrees	0.30
48	47 degrees	0.12
49	48 degrees	0.18
50	49 degrees	0.14
51	50 degrees	0.01
52	51 degrees	0.09
53	52 degrees	0.09
54	53 degrees	0.06
55	54 degrees	0.01
58	57 degrees	0.04
59	58 degrees	0.01
totals		186.74

Layer: [final9_DEM.wshed] in mapset [PERMANENT]
Title:
Mask: none

Window: north: 9005190.00 east: 394074.00
south: 8949366.00 west: 334938.00
res: 48.00 res: 48.00

Category		Acres	Hectares	Sq.mi
0	no data	784295.87	317400.19	1225.48
1		31432.64	12720.61	49.11
total		815728.51	330120.81	1274.60

ATTACHMENT G

DEMONSTRATION DESCRIPTIONS

[illegible]

Demo Items for Visualization of Multi-dimensional Data Sets

July 16, 1990

for Stardent's AVS Platform

Introduction

The use of Stardent's AVS applied to multi-dimensional image visualization is presented. This includes the use of application sub-systems, user interface menus, the modules palette and the network editor. Each example is grouped within the AVS subsystem used. The overall presentation is intended to demonstrate the *kind* of environment in which the basic visualization functions can be efficiently and extendibly hosted.

We have generated the input data sets by pre-processing steps but the intention is to eventually provide **modules** to perform the various required functions on demand by the user.

Network Editor

This sub-system is dedicated to building and editing networks of computational modules.

Example 1

An example providing selectable displays of three separate color slices of a cube and their RGB composite is given (3slicesrgb2). The data used is 15 bands (2.25-2.39 mic. wavelength) of a 255x255 pixel area of AVIRIS data taken over the Northern Grapevine mountains of Nevada. This example illustrates the concept of a user-defined data-flow network.

Example 2

The second example network (shadows2) shows some radar imagery in various forms: *direct*, *shadow map* and *bitmap outline*. This example demonstrates another use of a user-defined network, this time for images. Additionally, this example network contains a module generated by Vexcel.

Volume Viewer



The volume viewer uses cubes for input and produces various 3D renderings as well as some 2D displays. Currently volumes in AVS are limited to 255^3 bytes.

Example 3

This example shows the use of the Volume Viewer's *isosurface tiler* with AVIRIS data that has been compared to a laboratory spectra for a specific mineral. The data set used is a planimetric subset of the data used in Example 1 taken across the same # bands. The area of interest was decreased due to the performance constraints incurred by the additional bands as well as the ability to interactively manipulate a 3D rendering. The planimetric area is 100x100 centered on a known carbonate area.

In this case we start with **dolomite** since it is known through field analysis to exist in the area. This example illustrates the ability to view imaging spectrometer data in 3D using a spectral geological database.

An additional window provides a simultaneous display of the raw band imagery for the corresponding area. This window allows features in the 3D model to be compared with those found in the original imagery.

The user can interactively change the isosurface level, interpolation size, view angle and reference image band plane. An isosurface level at 250 (downsize=3) shows the likely presence of dolomite.

A number of other mineral-renderings are available for comparison with dolomite (calcite, chlorite). Interesting to note is the difference between dolomite and other substances at the very high confidence-isosurface level of 250!

Example 4

This example uses the same 100x100 area as the previous example but this time 15 bands are applied to a PCA transform to yield 15 principle component images.

The data that is input to this process is the residual data - that is the data that was actually rendered in 3D to indicate the presence of dolomite. The outlines found in successive components resemble features in the original and isosurface data.

Example 5

This example illustrates the use of multi-dimensional clustering using a minimization approach (AMOEBA). Here 10 bands (2.28-2.37 microm.) of original data in the 255x255 area of Example 1 are reduced to a 3 band RGB image where the Euclidean distance between original and resultant band pixels has been preserved. Again note the correspondence of features between this and other examples.

(Example 6)

This example shows again the use of the Volume Viewer's *isosurface tiler* with AVIRIS data that has been compared to a laboratory spectra for a specific mineral. This example looks at 15 bands that are centered not around carbonates but around the **gypsum** substance.

This example is identical in all other respects to Example 3.

(Example 7)

This example also shows the use of the Volume Viewer's *isosurface tiler* with AVIRIS data that has been compared to a laboratory spectra for a specific mineral. The data set used is a planimetric subset of the data used in Example 1 but taken across many bands.

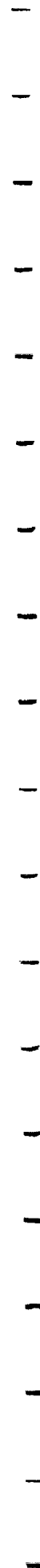
This example is identical in all other respects to Example 3.

(Example 8)

This example displays the results of a multispectral clustering process (AMOEBA) as applied to the 255x255 area of Example 1. The process maps R^n to R^3 while minimizing the Euclidean distance between pixels.

Here, we have filtered out noise bands from the original 210 band AVIRIS cube and then broken up this cube into 10 (more-or-less) evenly spaced sub-cubes. We have then applied the clustering to each of these N subcubes to produce N RGB images. Finally a composite RGB image was generated using the composites as input to the clustering.

Various bands of the resulting N+1 composites are then displayed. This example shows the use of minimization to reduce the dimensionality of multi-spectral data sets. It should be noted that features identified by the clustering correspond with those found by 3D rendering.



(Example 9)

This example is similar to that for the previous example except that the dimension reduction used was the Principle Components Analysis (PCA) transformation.

(Example 10)

Here we have generated residual volumes for 2 target substances (dolomite, calcite) in their respective strongest absorption feature bands regions. These cubes are then smoothed and applied to a PCA transformation. These images are then composited with an original band to form an RGB image indicating areas containing the targets. These results give a similar outline of features as previous examples. The area of interest is that of Example 1.

from: Michel
to: Matt, Woody

EOS DEMONSTRATIONS DESCRIPTION

From the EOS point of view there is 2 sets of demonstrations:

- . the demonstration panel
- . the LT demonstrations with contours lines and DLG data.

I will first describe the demonstration panel with the convention that a particular demonstration is identified by its row number and column number in the panel

. Row 1

. Column 1: Boulder cube. This cube contains 21 registered 512x512 images resampled at a pixel size of 25 meters (the projection used is UTM, the datum is NAD (North American Datum) 27 (adopted in 1927)) :

- . 1 multispectral SPOT image(3 layers) (3 spectral bands with nominal resolution of 20 meters, original size of 3000x3000 pixels, the SPOT instrument is also acquiring images in the panchromatic mode with a nominal pixel of 10 meters and a size of 6000x6000 pixels. For the panchromatic mode, the technology used is 4 1x1500 CCD matrices placed perpendicular to the flight and the integration lasts the time required to obtain a pixel of 10 meters in the flying direction) acquired with a viewing angle of 2 degrees (quasi vertical. The viewing direction is controlled by a mirror placed perpendicular to the flight direction and the range of possible angles is from -27 degrees to + 27 degrees). This image was purchased from SPOT image geometrically raw and geocoded by VEXCEL. More precisely, the mapping between the ground and the image was estimated using 6 ground control points (points identified on maps and in the images(the identification in the images was performed using LT) and V.Kratky modeling method (result RMS on 20 control points < 1 pixel) and the image was resampled at 25 meters pixel size using the previously estimated mapping and a Digital Elevation Model composed of 8 U.S.G.S 7.5 minute DEM (from 1:24000 maps) and 1 1degree DEM (from 1:100000 or 1:250000 maps). The 512x512 window was extracted from this ortho image.
- . 1 multispectral SPOT image (3 layers) acquired with a viewing angle of 27 degrees (oblique), same processing and characteristics as the previous image.

.1 geocoded THEMATIC MAPPER image (7 layers). This image was geocoded by EOSAT (the seller) with the following specifications: 30 meters resampled pixel size, projection UTM and NAD 27 datum. This image was registered to the ground without using a Digital Elevation Model but the geometry of the instrument (maximum viewing angle of ± 7 degrees at the extremities of each line), the nominal pixel size (28.5 meters) and the relative location of Boulder (near the Nadir) minimizes the effect of relief on the registration. THEMATIC MAPPER instrument is made of 16 detectors per spectral band (7 spectral bands) and 2 mirrors; 1 sweeping mirror perpendicular to the flight direction (in opposition with SPOT, this mirror is moving and doing the acquisition) and 1 mirror compensating for the satellite movement during 1 set of 16 lines acquisition (to preserve continuity of the images and perpendicularity with the flight direction).

.1 geocoded SEASAT image (1 layer) provided by JPL. I believe the pixel size in this image is 25 meters and no Digital Elevation Models were used in the geocoding process, explaining the distortion over the mountains. For more on radar, see Woody.

.1 contour lines image (1 layer). These contours are every 100 meters and were generated, as ASCII data, by CPS3 (Radian) using as input the already mentioned USGS DEMs and were rasterized to the image cube pixel size.

.1 hydrography image (1 layer) generated from 1:100000 (scale of the maps used) USGS DLG (Digital Line Graph). The data provided by USGS are the planimetric coordinates (UTM Easting and Northing) of points belonging to a particular class of map features, also provided with the data (and not used in our case) are some topological information about the relative location of specific features and their types.

.1 DEM image (1 layer). In this image the intensity is proportional to the height (the brighter, the higher). The DEM used are the ones previously described.

.1 classification image (1 layer). This image was generated by the AMOEBA program (provided by J.Bryan) with as input the 7 THEMATIC MAPPER images. For more on AMOEBA, see Woody.

.the first 3 Principal Components Analysis images (3 layers) of the THEMATIC MAPPER image. The Principal Components Analysis is a linear transformation, which can be intuitively imagined as looking for the principal axis of inertia of the object composed of the image intensities in N dimension (N is the number of spectral bands). For more on PCA, see Woody.

.Column 2: Boulder-RGB+. 2 color images are displayed. The background of the first one is made of a color composite of the vertical SPOT image geocoded to the UTM projection of the NAD 27 datum with a pixel size of 25 meters, the already described rasterized DLG is superimposed on it. The second image as the same SPOT image, but this time with the already described contour lines superimposed.

.Column 3: Boulder-Cut. An image cube containing 7 layers of the previously described Boulder image cube is inputted to the Region Of Interest program.

.Column 4: Boulder_Ccut. This demonstration shows examples of the Region Of Interest concept in color. All the images used come from the Boulder image cube, their pixel size is 25 meters and they are resampled to the NAD 27 UTM cartographic projection.

.Column 5: Boulder-Zcut. This demonstration shows the Region Of Interest concept with in addition the capability to use multiple resolution images. The SPOT image used is band #1 of the multispectral vertical SPOT image, It was geocoded to the NAD 27 UTM cartographic projection at a pixel size of 10 meters. The NHAP (National High Altitude Photography acquired and sold by USGS) image (original film scale ~ 1:50000) was scanned (in color) by USC (see Woody) with a spot size of approximately 2.5 meters. The mapping between the ground (1:24000 USGS maps) and this NHAP image was performed using FOTO-G, after acquisition of ground control points using LT. The NHAP image (red channel) was resampled using the 7.5 minute DEMs at 3 different pixel sizes: 2.5 , 5 and 10 meters.

.Column 6: Boulder-Prsp. This demonstration uses the STARDENT perspective view program. The DEM used is the 7.5 minute DEM mosaic, the DEM grid step is 50 meters and the image draped over the DEM is the THEMATIC MAPPER image from the BOULDER image cube. The triangle mesh was obtained by splitting each DEM square mesh into 2 triangles.

.Row 2

. Column 1: Boulder-Math. This Region Of Interest demonstration shows arithmetic operations that can be performed between layers of an image cube. the THEMATIC MAPPER image (band #1) is the same image used in the demonstration called Boulder-Cut.

. Column 2: Boulder-Cine. This image slicer demo shows another aspect of an image cube. Each layer consists of a perspective view of Boulder generated using the perspective view program. The image is SPOT band #1, the DEM grid step is 50 meters.

. Column 4: Nevada-Cube. Subset of an AVIRIS image (pixel size 20 meters, original image size 512x640, output image size 512x512). No geometric processing were applied to this image. In this particular image slicer demonstration, we have a pyramid of 2 resolutions for each image cube layer.

.Column 5: Nevada-RGB. Color composite of 3 bands of the AVIRIS cube the images were resampled (using bilinear interpolation) to become 1000x1000 (from 512x640)

.Column 6: Nevada-Cut1. a subset of 6 bands was selected from the AVIRIS image cube. Each selected layer was resampled (using bilinear interpolation) to become a 1000x1000 image. The seventh layer is the first Principal Component of the other 6 layers.

. Row 3

. Column 1: Nevada Cut2. A Principal Component Analysis was performed on the on the AVIRIS 6 layers image cube. The displayed images are the results of this PCA.

.Column 2: Radar-Brazeau. The images used at VEXCEL to present shape from

shading results. The Radar is STAR2 (resolution of 5 meters (along the flight) and 6 meters (in the range direction)). The used images are ground range images, the NHAP DEM was acquired on the DSR, the radar DEM was computed on the VAX, after acquisition on the DSR.

.Column 3: Shape-Shading. A subset of the Radar-Brazeau image cube is used to show in the animation loop the convergence of SHAPE FROM SHADING toward the original radar image. For more on shape from shading see Woody.

.Column 5: VenPersp. Perspective view of a shape from shading DEM with a Magellan image draped over it. For more information, see Woody

. Row 4

.Column 1: Growth. An example of Delaunay triangulation in real time. The planimetric coordinates are randomly assigned and the height is given by the formula written in the introductory text. This is a prototype of the future terrain window.

The EOS LT demonstrations are an example of virtual image cube and of the advantages of a good knowledge of the mapping between different sets of data.

. SPOT: to eliminate the rotation between the 2 images and possibly some pixel sizes differences, an affine transformation was applied to one of the image (SPOT 28) . The contour lines were computed from the USGS DEMs, and the UTM coordinates of each point along each contour line was transformed into a pair of image coordinates by application of the modified sensor model described earlier. The planimetric coordinates, associated with each DLG node, were used to interpolate (using bilinear interpolation), in the USGS DEMs , the corresponding height. Similarly to the contours lines, each set of DLG ground coordinates was transformed into a pair of image coordinates.

. RADAR: using a technic similar to the one described for SPOT, the contour lines (interval 250 meters) were projected into the pair of images.

General remarks:

Before, or during each demonstration, a text appears on the screen, explaining this demonstration. Also for the image slicer some labels are printed on the displayed cube layer, these are separate ASCII files called panel.dat or label.dat, if you do not like them modify them at your leisure. The only constraint is that the first line of text is the title, and is displayed with the strange font, your text will be centered and the "Press any mouse button to continue" message is fixed at window coordinates (y=900).

The directory organization and the list of interesting files can be found in the ~demo/dui/data.menu file.

ATTACHMENT H

**REPRINTS/COPIES OF TECHNICAL PAPERS
ABOUT
THE EOS WORKSTATION**

PRECEDING PAGE BLANK NOT FILMED

REQUIREMENTS FOR AN EOS-ORIENTED WORKSTATION

W Kober, J Thomas, D Meyer, F Leberl

J Cimino

VEXCEL Corp.
2905 Wilderness Pl.
Boulder, CO 80301, USA
Tel: 303/444-0094

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109, USA
Tel: 818/354-8225

ABSTRACT

We discuss the initial efforts for the determination of functional and performance requirements of a workstation specifically directed toward scientific users of the proposed NASA Earth Observing System (EOS) Information System.

This vision of an EOS information system which coordinates the distribution of EOS data suggests a requirement for a workstation which allows the individual scientific user convenient access and interaction with such a system (see Figs. 1 and 2). For example, users of EOS information will have requirements and access patterns that are typical of their fields. Some users will require wider coverage with low resolution data and others will need smaller coverage with higher resolutions.

A composite of the many types of anticipated user data requirements includes: catalog searches by regions, wavelength, resolution, or times; browsing of low resolution data; scanning data of given regions along different "dimensions" of imaging parameters; time series analyses of low and high resolution data.

Present image processing systems often have the capacity to store large amounts of data, but do not have capabilities for conveniently accessing data in the modes described above.

Moreover, it is important that such a workstation should efficiently and conveniently execute certain data processing functions which are expected to be common to EOS-oriented scientific users.

Automated processing requirements include capabilities for geometric and radiometric manipulation of large data sets. Among these functions are dead-reckoning of image data, coregistration of dissimilar imagery, analysis of image content, and a range of tools for visualization of multiple imagery sets.

Commercial image processing systems have applications functions that are oriented toward general image processing, rather than on the functions mentioned above for the specific sensors that are planned for EOS such as MODIS, HIRIS, and SAR (Table 1).

Such a set of functions should contain tools which allow the user to conveniently browse and select data online, and to organize and examine the received data. It is expected that the full dataset will generally be transferred offline using media such as optical disk.

1. INTRODUCTION

A major thrust of the Earth sciences in the 1990's and beyond will be the systematic, unified investigation of dynamic global-scale processes and of changing regional phenomena. The NASA Earth Observing System (EOS) is intended to provide the capability to scientifically monitor these global scale, dynamic processes over long periods of time. The systematic study of these global processes will require the collection and organization of very large quantities of raw and processed sensor data, collateral data, and address a great range of physical parameters [Ref.4].

In the past, programs dealing with Earth observation were more specifically sensor based. Large quantities of data were created and archived, but the majority was not used or even effectively cataloged. Because of these previous experiences, NASA's planning for the EOS concept emphasizes its role as an information system, of which the multiple sensor platforms are only one centralized data centers, which are to be readily accessible by scientific users.

These data issues lead naturally to a vital role for a Geographic Information System (GIS), ie. a spatial database. Its role is discussed in more detail in section 3.4.

The following section briefly outlines the methodology for specifying the requirements for such a system. The driving user requirements are identified both for functions and performance characteristics.

2. METHODOLOGY FOR DEVELOPMENT OF A SPECIFICATION

At this time, a detailed specification of all the EOS-type image processing functions has not yet been defined. The methodology for determining the requirements for an EOS-oriented workstation and formulating preliminary design recommendations involves first understanding the typical operational scenarios for users in relevant scientific disciplines. These scenarios are used to derive the workload and functional requirements for the workstation. This process is illustrated in Fig. 3.

The types of scientific disciplines and backgrounds which are expected to use such a workstation include [Ref.3]:

- o global hydrologic cycle
- o global biogeochemical cycles
- o biological oceanography
- o inland aquatic resources and biogeochemical cycles
- o forest environments
- o land biology
- o tropospheric chemistry
- o geology
- o interior of the earth
- o oceanic transport
- o polar glaciology
- o sea ice
- o tropospheric science
- o middle atmosphere science
- o aeronomy

Although numerous sensor types are under consideration for EOS platforms, the most important of these are expected to be MODIS, HIRIS, and SAR (see Table 1).

The typical analysis workloads involve various data types consisting of sensor, calibration, and ground truth data, as well as non-EOS collateral data. The volumes and access patterns of the various user types for browsing and down-loading data will drive the workload specifications.

The functional requirements will be the specifications for image processing and display, data analysis, and database management. Functional performance, such as accuracy, will be included. Accurate models and data rates of the relevant EOS sensors will be required for creating these specifications.

EOS-type data analysis is expected to be region driven, data driven, or problem driven. Therefore, the workstation must have the capability to support user access to data using multiple keys for searching the database (see section 3.4).

The workstation's execution of these functions on the workloads will be constrained by the specification of approximate response time or throughput requirements for functions, and capacity requirements for data storage.

These specifications for workloads, functions and performance will be used to derive recommendations for software, hardware, and system configuration.

The three image processing functions that are considered of utmost importance and whose performance requirements drive the system design are geocoding, coregistration, and visualization of multiple bands of imagery. These functions are discussed in section 3.5.

For an EOS-specific workstation to be useful, it must satisfy the throughput requirements of prospective users. For enhanced capabilities, both functional and performance, it is intended that the basic workstation configuration will be compatible with optional "add-on" hardware modules.

A top-level design for the basic workstation configuration is under development as a result of the ongoing specification of requirements. Some discussion of preliminary conclusions of the present study follows in the next sections.

3. LOGICAL ORGANIZATION OF THE WORKSTATION

3.1 General

One unique aspect of an EOS-relevant workstation is its reliance on a GIS Management System (GISMS) for coordinating all relational aspects of processing. For example, although raster images are not physically stored in the GIS, the boundaries of the image coverage areas are stored there. Pointers to the appropriate images in the image database are also stored there.

The overall organization of the EOS workstation software from the standpoints of flow of control and flow of data is shown in Fig. 4. The four main functional blocks of modules are the user interface, the applications executive, the block of applications software, and the GISMS.

3.2 User Interface

The user interface coordinates the user's physical access to the workstation. The devices that it manages for input and output will include the displays, the keyboard, joystick and mouse devices, the printer and image hardcopy device.

The user interface also provides syntactical checking of all lexical input commands, and contains a window manager for parsing iconic and graphical inputs.

3.3 Applications Executive

The applications executive accepts syntactically correct inputs from the user interface and initiates the necessary commands for the activation and execution of applications software. Its logical structure is a collection of command files and a selection module for choosing the appropriate command file.

Each command file contains logic for insuring that the applications software for which it is responsible has access to the required inputs. These may be user-provided inputs, as well as internally stored data types such as image, calibration, geographic, and others.

Obtaining the necessary inputs will generally require formulating queries to one of the databases. These may be generated directly by the user or indirectly by some software module. Because of the central role of the GISMS, any query that does not explicitly reference a particular data item will be referenced using the GISMS even though that data item may likely not reside in the GIS. The command files in the applications executive will coordinate the transfer of data products to and from the user interface and the databases.

These command files also contain logic for semantic checking of commands. For example, if a request for data has been formulated and these data are not available, then this information is passed on to the user and the appropriate applications module is not activated.

3.4 GIS Management System (GISMS)

The GISMS controls access to the three main databases of the workstation system: the GIS or spatial database, the image database, and the text database. This coordination will allow the formulation of complex relational queries which combine information from all the databases.

As discussed above, the spatial database contains geographically oriented pointers to other databases, as well as geographically indexed collateral data. The image database contains all sensor images.

The text database contains all textual annotations of data, such as calibration information and processing history. It can also be used to organize documentation of research such as reports and papers.

3.5 Applications Software

The Applications Software block will contain all of the image processing and data reduction functions. Three applications in particular are expected to drive the computational capabilities of the workstation. These are geocoding, coregistration, and visualization of multi-dimensional data sets. These three functions are discussed in subsections 3.5.1-3.5.3.

Some capability for atmospheric calibration will also be included. However, this particular application will not be considered as important for driving the specifications as the three functions mentioned above.

The Applications Software block will have the capability of containing multiple copies of certain selected applications modules according to data type. For example, certain numerically intensive routines will be realized in software as separate copies for efficiently dealing with inputs in various integer or real data formats.

3.5.1 Geocoding

Geocoding is the process of assigning geographical coordinates to pixels in an image. Geocoding allows the presentation of an image to be viewed as a map, rather than as the particular geometric transformation of the sensor that is involved.

This is particularly useful in the case of SAR imagery, where the near range portion of the image tends to be rather compressed.

However, image registration with a map allows not only a more natural presentation, but also makes it possible to readily cross-reference and access ground truth and other ancillary data stored in a spatial database. In fact, such geographically based organization of imagery allows efficient ways of selecting and examining archived imagery.

Mathematically, geocoding is the determination of a geographical pre-image of the sensor transformation for each image pixel. Geocoding first requires relating the sensor coordinate system to an Earth coordinate system. Dead reckoning, tracking, and ground control can be used to obtain ephemeris estimates. Platform attitude sensors or ground control are generally used for attitude information, although SAR phase history may also be used [Ref.1].

After relating the sensor and Earth coordinate systems, a reference geoid or terrain model is used for determining an intersection with the locus of points constituting the sensor transformation's pre-image.

Although it is anticipated that imagery will be geocoded at centralized EOS facilities, the EOS workstation must have a capability to also accomplish this task to potentially higher accuracies. Because the different sensor types employ differing geometric transformations, a separate geocoding procedure must be used with each sensor type.

3.5.2 Coregistration

Intuitively, coregistration is the process of spatially transforming or "warping" different images of the same scene so that when they are stacked, the overlaying pixels correspond to the same ground resolution element.

The accuracies of geocoding and rectification procedures are limited by the accuracies of their input reference data, such as ephemeris, ground control, and terrain models. These accuracies may not be sufficient to achieve the coregistration of independently geocoded images.

The joint use and application of multiple-sensor image data sets is necessary for achieving the full potential of remote sensing. Therefore, a capability for coregistration distinct from geocoding is required.

Coregistration is generally a difficult problem because of the dissimilarities in images. A recent review in [Ref.2] enumerated 90 references. For example, some of the variables which can potentially contribute to dissimilarities in imagery are differences in:

- sensor type
- wavelength
- incidence angle
- polarization
- illumination changes
- scene content changes
- sensor position changes

The workstation will be used to support both automatic and interactive coregistration of dissimilar source imagery. However, because such automatic methods are not always reliable, the workstation design considerations will be driven by the need to efficiently support the interactive approach. Interactive techniques can also be used

for correcting registrations produced by automated algorithms.

An example of the interactive coregistration of aircraft and SIR-B satellite SAR imagery, taken from [Ref.8], is shown in Fig. 5-8. Matching points were obtained interactively to create a deformation grid for the image that is to be warped. Then this image is resampled as defined by the deformation grid.

One of the sources of error for interactive registration is the small residual pixel error in specifying the initial match points. The EOS workstation will have the capability to approximately specify such points for subsequent refinement. A background software process will subsequently refine the locations of these points using the computed intersections of refined edges. An example of this capability is shown in Fig. 9 and 10. These figures show the complexities involved in coregistration arising from image dissimilarities.

3.5.3 Data Product Visualization

It has been said that science is generating data with 21st century technology, but is using 19th technology to understand this data [Ref.6].

An important feature of the workstation for scientific usage of EOS data will be the capability to visualize multiple data products efficiently. The usual presentation aids available from image processing techniques will include IHS-RGB transformations [Ref.7], digital elevation models, and stereoscopic, orthographic, and perspective views. Examples of perspective view presentations can be found in Fig. 11-13.

The large number of spectral bands of overlapping EOS-type imagery increases the "dimensionality" of the data set so that there is a potential problem for effectively viewing this data. Rather than trying to find a simultaneous visual presentation of this higher dimensional data set, the approach taken for the EOS workstation involves an effective set of tools for creating and viewing sequences of lower dimensional processed images. Such an image sequence would "freeze" spatial dimensions and show variations arising from other parameters. Shown at near-video rates, such a sequence can allow the user to "navigate through the parameters" of the data set.

What is required is a convenient interactive facility for specifying such a sequence of lower dimensional images as a function of chosen parameters. The workstation will allow such interactive parameter specification using a joystick. The actual processing and selection of the imagery will be a background processing activity, but the specification process will be real-time. An example of such a presentation is in Fig. 14, where the desired parameter has been encoded as false-height disparity and is visualized using a perspective view.

This "joysticking" capability will also be applicable for specifying viewpoints for generating the perspective views of terrain model data discussed above. In this way, a "fly-by" sequence can also be interactively specified. Again, the actual computations of the perspective views will be generated as background processing. Once completed, the sequence of computed "fly-by" views can be viewed at near-video rates.

3.5.4 Text Processing

The capability for creating research documentation should also be hosted on the workstation. Recently, research into "hypertext" has prompted the development of research tools which combine word-processing and database manipulation [Ref.5]. Such tools are used for organizing a writing environment which goes beyond the lexical features of ordinary word-processors.

Such an environment allows a user to outline, structure, link, search, and display text fragments according to some conceptual structure. An analogy can be made with a collection of topical notecards used in the organization of a research paper. A recent commercial example of such a hypertext facilities is Apple Macintosh's Hypercard.

Eventually, this workstation should have the capability to integrate such a hypertext facility into its operation. This will more fully automate the process of merging the data and text that is so characteristic of writing research papers.

4. SUMMARY

The EOS workstation described above will allow EOS-era researchers to more fully realize the potential benefits of working with large sets of multi-spectral data. There is a high expectation for a dramatic increase in user efficiency for choosing, organizing, manipulating, and visualizing this EOS data.

5. REFERENCES

1. Curlander J., "Utilization of Spaceborne SAR Data for Mapping," IEEE Trans. Geo. Rem. Sens., GE-22, #2, Mar. 1984.
2. Leberl, F., "Review of Basic Techniques for Multi-Sensor and Image Time Series Analysis," VEXCEL Technical Report #100986 (to European Space Agency), Boulder, CO 80201, Oct. 1986.
3. Earth Observing System, Science and Mission Requirements Working Group Report, Vol. I, NASA, 1988.
4. From Pattern to Process: The Strategy of the Earth Observing System, EOS Science Steering Committee Report Vol. II NASA, 1988.
5. Conklin, J., "Hypertext: An Introduction and Survey," IEEE Computer, Sept. 1987.
6. Wolff, R., "The Visualization Challenge in the Physical Sciences," Computers in Science, Jan/Feb 1988.
7. Haydn, R. et al, "Application of the IHS Color Transform to the Processing of Multisensor Data and Image Enhancement," Int. Symp. on Rem. Sens. of Arid and Semi-Arid Lands, Cairo, Egypt, Jan, 1982.
8. Thomas, J. et al., "Concepts for Processing and Analyzing of Multiple SAR and Landsat Images," Proc. SPIE Orlando '88, 1988.

SAR

Sensor Type: Synthetic Aperture Radar
 Incidence Angles: 15 deg. - 55 deg.
 Frequencies: L, C, X Bands; L, C Quadpole
 Polarization: HH, VV
 Height: 824 km.
 Azimuth Angle: - + 60 deg.
 Resolution Modes:
 Local High Res
 Swath: 30 - 50 km.
 Res: 20 - 30 m.
 Regional Mapping
 Swath: 100 - 200 km.
 Res: 50 - 100 m.
 Global Mapping
 Swath: max. 400 km.
 Res: 200 - 500 m.

HIRIS

Sensor Type: High Resolution Imaging
 Spectrometer
 Height: 824 km.
 Resolution: 30 m.
 Swath: 30 km.
 Spectral Range: 0.4 - 2.5 μ m.
 Number of Channels: 200

MODIS-T

Sensor Type: Medium Resolution Imaging
 Spectrometer
 Height: 705 km.
 Resolution: 1000 m.
 Swath: 1513 km.
 Spectral Range: 410 - 1040 nm.
 Number of Channels: 64

MODIS-N

Sensor Type: Medium Resolution Imaging
 Spectrometer
 Height: 705 km.
 Resolution: 500 m.
 Swath: 1513 km.
 Spectral Range: 470 - 2130 nm.
 Number of Channels: 25

TABLE 1

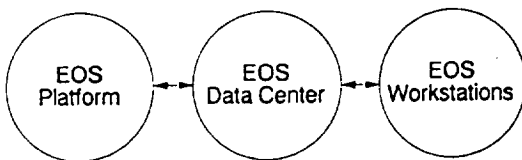


Figure 1. The EOS System

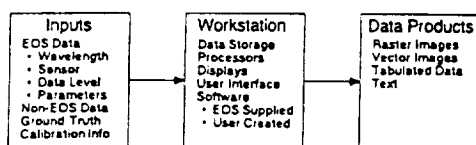


Figure 2. User View of Workstation

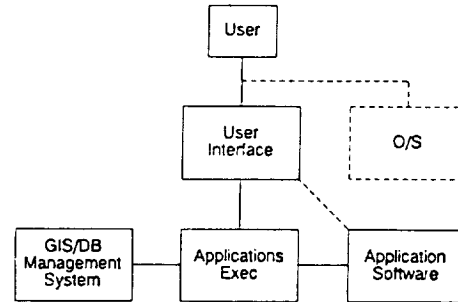


Figure 3. Major Software Sections

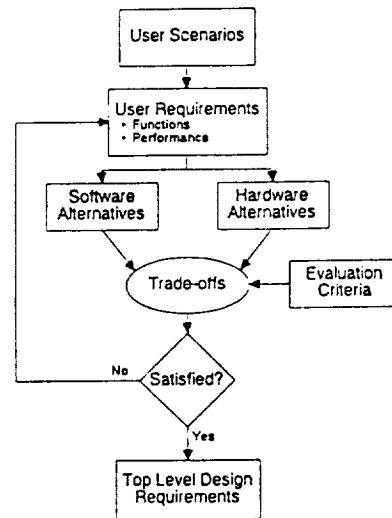


Figure 4. Workstation Design Process

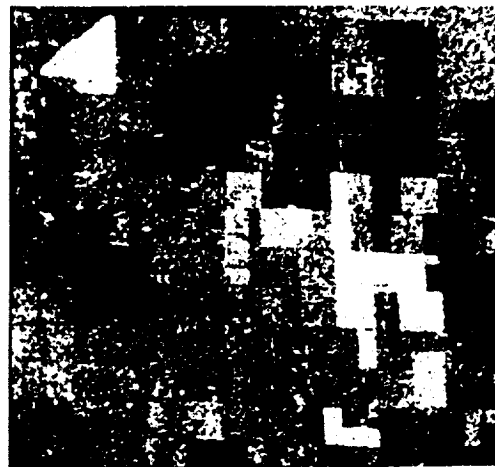


Figure 5. JPL Aircraft-SAR Image #1

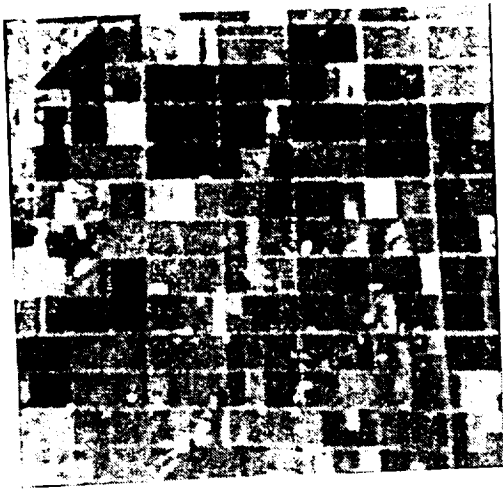


Figure 6. Landsat Image

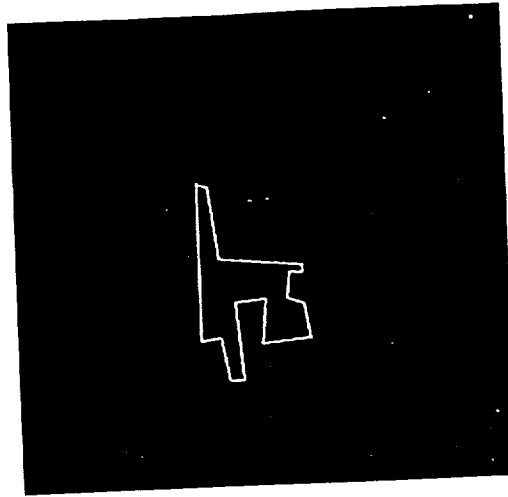


Figure 9. Operator selected edges of a feature in Figure 5.

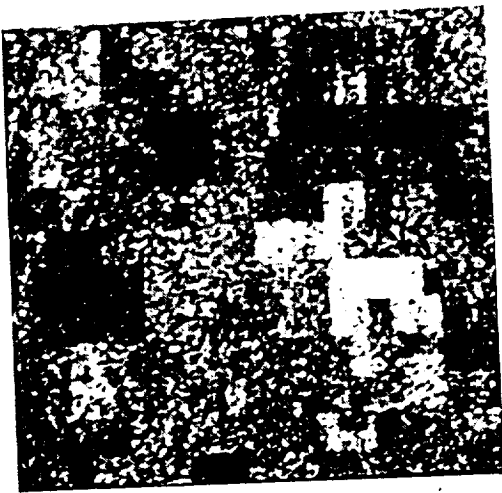


Figure 7. SIR-B SAR Image

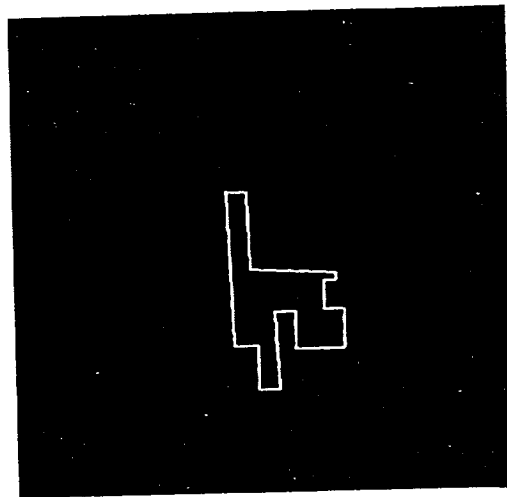


Figure 10. Automatically corrected edges of the same feature as in Figure 9.

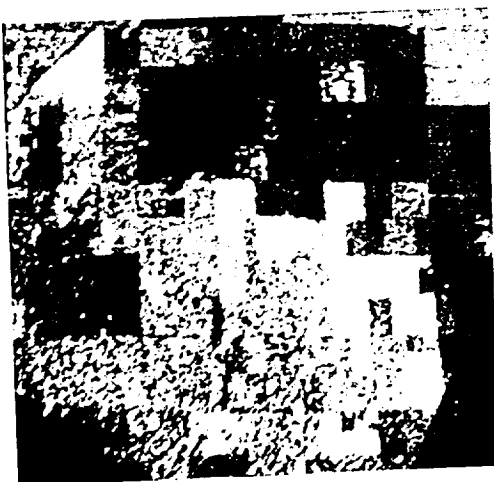


Figure 8. JPL Aircraft-SAR Image #2

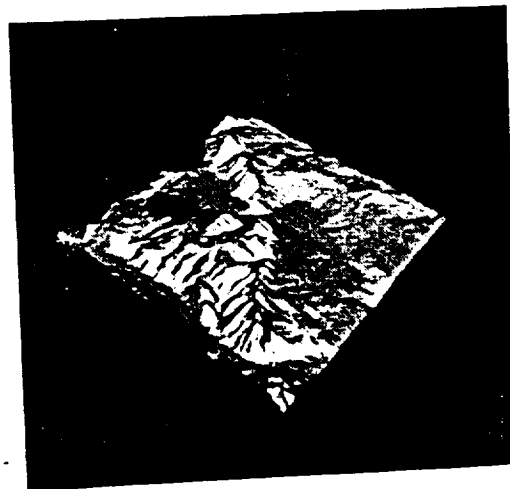


Figure 11. Perspective View

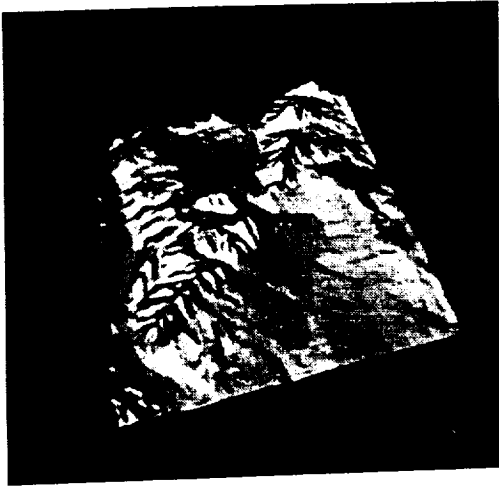


Figure 12. Perspective View

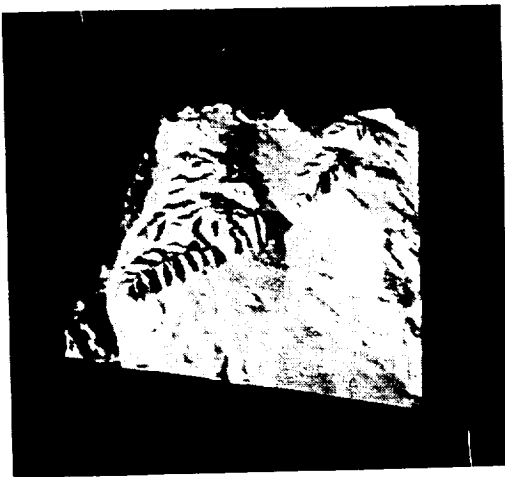


Figure 13. Perspective View

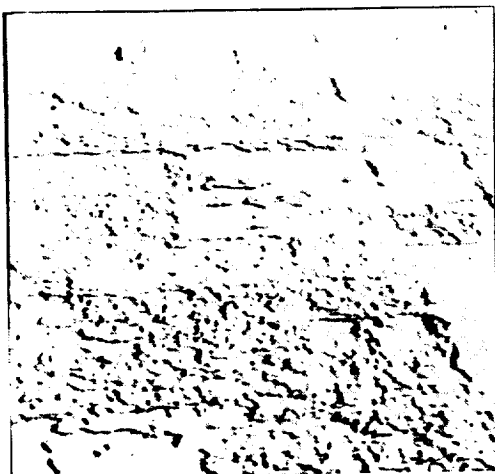


Figure 14. The ratio between Figure 5 and Figure 6 encoded as height and presented as a perspective view.

VISUALIZATION ENVIRONMENT FOR IMAGING SPECTROMETRY*

+K. Maurice
+W. Kober
*F. Kruse

+ VEXCEL Corporation
2477 55th Street
Boulder, CO 80301 USA

* Center for the Study of Earth from Space
University of Colorado, Boulder
Boulder, CO 80301 USA

1. ABSTRACT

Initial work is presented using volumetric rendering techniques for visualizing high-dimensional imaging spectrometry data sets. The presentations discussed can be used for presenting residual values with respect to laboratory spectral prototypes for specific minerals.

Preliminary results show promise for the use of such a visualization technique to complement automated classification procedures, both for viewing final results as well as for examining data during intermediate stages of analysis.

A brief discussion of the some of the issues involved and difficulties encountered in using volumetric rendering for viewing such data sets follows. Finally, recommendations are given for using and enhancing a volumetric rendering environment in the context of viewing imaging spectrometry data.

2. INTRODUCTION

Scientists have worked with multispectral digital images and display techniques for many years. However, there is little experience with very high dimensional data sets. Previous studies of data from the Airborne Imaging Spectrometer (AIS) and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) have shown that the effective analysis of such multi-dimensional imaging spectrometer data is greatly enhanced by "visualizations" of the data set.

In particular, it would be very helpful to simultaneously visualize information in all the bands of a data set. The need for such a capability may become more acute in the case of data from NASA's future EOS instruments, HIRIS AND MODIS. At present, the technology to realize such a multi-dimensional "mindset" has not been realized.

Clearly, visualization of data for physical interpretation is important. However, another important role for visualization is the inspection of intermediate stages of data pre-processing.

This paper concerns the initial portion of a research effort defining and

*Presented at Eighth Thematic Conference on Geologic Remote Sensing, Denver, CO, USA, April 29-May 1, 1991.

exploring a novel approach to this visualization problem and documenting its feasibility. The initial approach included an exploration of the use of volumetric rendering for visualizing high dimensional multispectral imagery. The paradigm uses laboratory prototypes of the spectral responses of specific minerals to be compared in the sense of residuals to the actual sensed multi-band, reflectances on a pixel-by-pixel basis.

Previous efforts in this direction have been able to display the results of comparisons for individual pixels. However, the use of volumetric and iso-surface rendering allows a visual examination to be made simultaneously for an entire region.

The capability to view an entire region of comparisons (residuals) should complement the use of automated classification methods, and is not by any means meant as a replacement. This capacity to see patterns at a glance will be made more meaningful by adroit use of annotations for physically meaningful quantities. In particular, a composite "cube" with map and projected view data added as layers will allow additional geometrical and morphological context to be applied to visualization of spectral data.

3. BACKGROUND OF IMAGING SPECTROMETRY

3.1 GENERAL ISSUES

In the 1980s, imaging spectrometry has been demonstrated as a viable concept has been successfully used as a key component of broad-ranging geologic and environmental studies. Imaging spectrometers have made possible the direct identification of Earth surface materials based on their diagnostic spectral characteristics measured in narrow contiguous bands. The advent of imaging spectrometers marks a change in direction from qualitative, empirically based remote sensing to deterministic, quantitative measurements. This change realizes an important goal of all terrestrial remotely acquired measurements, which is to quantify the physical parameters related to patterns and processes at the Earth's surface.

During the mid-1990s, the first satellite imaging spectrometer called the "High Resolution Imaging Spectrometer" (HIRIS) will be launched as a facility instrument on the Earth Observing System (EOS). HIRIS will provide simultaneous spectral measurements in 192 bands between .4 and 2.5 μm with a regional field of view [NASA,87], at a raw data rate of 512 Mbs [Goetz and Herring,89].

The HIRIS will use two dimensional area arrays to image an approximately 30 km swath. Another instrument called the Moderate Resolution Imaging Spectrometer (MODIS) will image an 1800 km swath with 1 km resolution in up to 64 channels [NASA,86]. These two instruments will present new opportunities for synergism between the high resolution HIRIS instrument having limited spatial coverage to be used in a targeting mode and the lower resolution MODIS instrument having repeat global coverage.

The particular imagery set used in VEXCEL's research was a multispectral data cube obtained from the Airborne Visible/Infrared Imaging Spectrometer

(AVIRIS). This sensor has 224 channels, each approximately of 10 nm bandwidth, operating in the $[.41, 2.45]$ μm wavelength range. The data set used in the present effort was spectrally resampled to 210 channels by JPL. The AVIRIS sensor was flown aboard the NASA ER-2 aircraft at an altitude of approximately 20 km, with an instantaneous field of view of 20 m and a swath of about 10 km which is scanned as 614 pixels in the direction perpendicular to the direction of flight.

The imaged site was the Northern Grapevine Mountains in northern Death Valley, California and Nevada during May 1989 as part of the AVIRIS evaluation program. The data were converted to reflectance using ground spectra and a technique called "empirical line calibration" [Kruse,90a].

The relevance of this AVIRIS data set is as a prototype for the future NASA EOS-era imaging spectrometers such as HIRIS, MODIS-T, and MODIS-N. Moreover, the present analysis techniques based on laboratory prototypes [Kruse,90a] provide the motivation for the visualization paradigms discussed in this report.

The scientific scenarios for use of HIRIS image cubes are expected to include [Dozier,88]:

- o mineralization in rocks and soils,
- o algal pigments in oceans and inland waters,
- o plant canopy biochemistry,
- o composition of atmospheric aerosols,
- o grain sizes and impurity absorptions of snow.

The intent is that the HIRIS will be used as a targeting instrument together with the MODIS instrument in a multi-stage sampling strategy [Goetz and Herring,89].

The complementary EOS-era instruments will be the Moderate Resolution Imaging Spectrometers, MODIS-T (tilt) and MODIS-N (nadir). The MODIS instruments are intended to be used for obtaining continuous coverage over at least 10 years. The scientific application areas, involving the monitoring of global dynamics and processes, will be the same as described above for HIRIS, but in this case will support daily, weekly, monthly, and interannual analyses of variations [Salamonson,88].

3.2 GEOLOGICAL APPLICATIONS

One demonstrated application of imaging spectrometer data is for detailed mineral mapping for mineral exploration using prototype or library spectra. The following is a description of a typical AVIRIS analysis scenario for geologic targets [Kruse,90a].

AVIRIS data acquired during May 1989 were analyzed for the Northern Grapevine Mountains site. Field spectra of known targets were used to calibrate the

data to reflectance with the empirical line method [Roberts et al,85]. Spectral signatures were extracted using interactive display and analysis software allowing identification of individual minerals.

The short-wave infrared data from 2.0 to 2.5 μm was used to identify and map the distribution of clay and carbonate minerals. The visible and short wavelength infrared portions of the spectrum ([.4,1.2] μm) allowed identification and mapping of iron oxide minerals. Fig. 1 shows spectra extracted from the AVIRIS data for known occurrences of calcite and dolomite.

Comparison of the shapes and positions of the absorption features to laboratory spectra makes positive identification of the three minerals possible, as seen in Fig. 1. The AVIRIS data not only allow identification of the carbonate-group minerals, but allow identification of the individual species (calcite and dolomite) based upon a 20 nm (2 channel) difference between the position of the main absorption feature, 2.34 μm vs 2.32 μm .

4. VOLUMETRIC RENDERING FOR VISUALIZING RESIDUALS

4.1. BRIEF OVERVIEW OF VOLUMETRIC RENDERING

Volumetric rendering is a technology for directly viewing entire volumes of data. Among the earliest motivations for such visual presentations was the requirement to view 3-D medical tomographic data. Some of the older methods for rendering volumes involve either extracting surfaces between materials directly or creating polygons at each pixel.

The direct extraction method involves the creation of contours, either manually or automatically, in each of the "slices" of the data set. Contours in adjacent slices are then connected using triangular strips or higher order patches. Locally ambiguous branchings often make this procedure difficult.

Three main methods are used for polygon creation: cuberille, "marching cubes", and "dividing cubes". The cuberille method views each voxel as a small cube whose faces are square polygons. Adjacent cubes are merged into an octree structure. The method involves setting a threshold for transition between materials and creating a binary volume for a particular material. The "marching cubes" method samples values at the vertices of each cube and thereby estimates surfaces cutting through a cube. This is a method also used for the creation of iso-surfaces of volumetrically defined scalar values.

Newer methods involve additive re-projection techniques as described in [Levoy,88,90], [Sabella, 88], [Upson and Keeler,88]. These ray-tracing methods average the volume intensities along parallel rays, thus simulating x-ray imagery.

Another feature involves source attenuation. There is a source strength at each voxel which is attenuated by the opacities of materials. This allows obscuration as well as attenuation. Depth cueing can even be incorporated using ray tracing. A ray is traced until a surface is reached. Shading and intensity values inversely proportional to viewing distance help create a depth effect.

Material percentages can be determined using probabilistic classifications. Material percentages allow the calculation of color and opacity of a volume as the sum of the products of (% material) x (material color) and the sum of products of (% material) x (material opacity).

Materials boundaries are detected using materials gradient operations, resulting in surface strength estimates which are based on the magnitudes of these materials gradients. Surface shading calculations are also based on these gradients.

A shaded color volume is composed of components emitted, scattered, and reflected. The emitted component is proportional to the amount of luminous material in a voxel.

The reflected component is a function of:

- o position/color of light source,
- o position of viewer,
- o surface normal,
- o surface strength,
- o color of the volume.

Essentially this approach to rendering involves the evaluation of k samples along each parallel ray, one ray per pixel. For each sample, the calculation involves the eight neighboring voxel values which are trilinearly interpolated. A color composite is formed by taking samples along each ray from back to front.

For each sample, the color leaving the sample location is a function of the color entering the voxel and the color and opacity within, using the transparency formula [Levoy,88]:

$$c_{out,\lambda}(u) = c_{in,\lambda}(u) (1-\alpha(x)) + c_{\lambda}(x)\alpha(x) \quad (1)$$

where: u = voxel position

x = distance along ray

α = opacity

λ = color, ie. R,G, or B

4.2 APPLICABILITY OF VOLUMETRIC RENDERING

The problem of visualizing the entirety of a data cube such as from the AVIRIS or the future MODIS and HIRIS sensors is appreciated by researchers using high-dimensional imaging spectrometry data. A display of the entire data set can be created which is colorful and visually striking. However, it is perhaps less clear how to make such a display useful for scientific analyses

of surface phenomena.

Previous approaches to visualizing multispectral image data with fewer wavelength bands have included the creation of color composites.

It is clear that successful visual presentations for analyzing high-dimensional multispectral data require selective presentations that are relevant to the presence of particular substances. This is the approach of using laboratory prototypes of spectra for specific substances, and visualization using this approach is the basis for the present effort.

As discussed in section 3, the identification and analysis of substances from multi-spectral imagery requires knowledge of prototype spectral signatures of these substances. Comparisons of such spectral data with prototype signatures for entire planimetric regions is more difficult. For example, averaging of the spectral values for an entire region is one method which allows the use of the previous display method (see Fig. 1) at the expense of some loss of information.

Moreover, the definition of a "good fit" of a measured spectral curve to a laboratory prototype is not always straightforward and can be somewhat ambiguous analytically. Consequently, ordinary clustering schemes which depend on such analytic metrics could encounter ambiguous situations.

Therefore, there is a requirement for the capability to see the correspondence of an entire region with a given prototype spectral curve, not just an individual pixel. Moreover, such a method should allow a presentation of spectral curve disparities that is robust with respect to these analytic ambiguities for spectral curve similarity.

Volumetric rendering satisfies both of these requirements, and has promise as a tool for interactive clustering. Moreover, volumetric rendering can be useful for making comparisons of the visual spectra of derived inverse ground modeling results with the actual sensed results.

Because of limits on spatial resolution, multispectral imaging sensors often capture image data which contains "mixed pixels". These mixed pixels represent multiple populations to be classified which are imaged together in the same pixel. Clearly, the probability of such mixed pixels increases as the sensor's spatial resolution decreases.

Considerable effort is being expended for the development of automated methods of decomposing multispectral data sets involving pixels of mixed populations [Adams and Smith,86], [Mustard and Pieters,87a,87b], [Singer and McCord,79], [Smith and Adams,85], [Smith et al,85], [Boardman,90]. Such methods represent solving difficult "inverse" problems.

For the present effort, only iso-surfaces have been created and used for visualization. However, it is hoped that in the future both iso-surfaces and newer, more faithful volumetric rendering techniques can be used together to visualize multispectral data cubes. The use of truer volumetric rendering will allow a fuller examination of the data, while the iso-surfaces will provide noise smoothing and will give a geometrically oriented depth context.

5. ADDITIONAL DATA CONDITIONING

Data conditioning can have two roles. The first is concerned with the removal of noise. The second is concerned with the ability to cue potentially interesting planimetric regions corresponding to selected minerals.

5.1 NOISE REMOVAL AND SMOOTHING

A fundamental pre-processing step normalizing all laboratory and data spectra is required prior to making any comparisons. This technique, called "continuum removal", is done in the calibration process described in [Kruse,90b].

After data calibration there remains a noise component in the data set. Such noise usually contains both random and fixed pattern types. An interesting example of the latter is the recent discovery of the existence of sensor-induced striping in AVIRIS imagery [Rose,89]. No attempt was made to ascertain the degree of this effect in the present data set.

Many methods for reducing random noise could be used. In the present study, a method is incorporated which provides both some degree of smoothing for noise reduction as well as data compression [Mazer et al,87,88]. The use of this method considerably aids the visual clarity of the appearance of the 3-D datacube, and speeds up the automated analyses.

Each reference spectrum is encoded by finding its mean and determining whether each point in the spectrum is above or below the mean. The spectrum is stored as a sequence of byte vlaues with each byte representing a point in the spectrum. If a point is above or equal to the mean it is set to 1 and if a point is below the mean it is set to 0. The same is done for the slopes about zero and appended to the previous byte sequence.

The spectrum for each pixel in the image is encoded in the same manner and compared to the reference spectrum using a bitwise exclusive OR. The exclusive OR determines points where the encoded spectra do not match.

For automated matching, a tolerance is used to determine how many points in the spectra must match in order to classify that pixel as being a match to the reference. A pixel exceeding this tolerance is given the maximum residual, while the others are given an inverted, scaled, residual absolute value. For example, a value of 255 corresponds to a perfect match while a value of 0 corresponds to the highest degree of mismatch.

5.2 REGION CUEING

Binary encoding is a fast and accurate technique for identifying minerals with distinct absorption bands because it is sensitive to band positions and insensitive to albedo (brightness) variation. Potentially, an expert system which analyzes binary encoded spectral data could be used together with volumetric viewing. As an example, the expert system could cue regions which potentially correspond to selcted minerals. Such regions cued by an expert system could then be viewed volumetrically.

6. RESULTS

Fig. 2 is an actual iso-surface rendering of residuals formed from calibrated AVIRIS data and the laboratory spectral prototype for dolomite. Fig. 3 shows the classification map produced by supervised classification [Kruse,90a] using the same laboratory spectral prototype data for the same region. The close similarity of the dolomite regions in Fig. 2 and Fig. 3 suggests the potential synergy between automated data analysis and volumetric visualization of the corresponding data set.

Finally, Fig. 4 shows a composite of the more traditional visualizations of this same region. Included are principal components, individual band planes, and residual plots for single pixels.

7. PROBLEMS ENCOUNTERED

Some of the problems observed in the course of experimenting with volumetric rendering included:

- o Data problems,
 - noisy spectral data,
 - calibration errors distorting the magnitude of absorption spectra,
- o 3-D visualization problems,
 - ambiguities of selecting opacities for residual levels,
 - ambiguities of selecting iso-surface levels,
 - lack of transparency of iso-surfaces,
 - lack of color coding for iso-surfaces,
 - inability to render more than one iso-surface,
 - inability to see rendered objects in stereo,
 - inability to selectively "peel away" regions to examine other regions that are occluded or obscured,
 - inability to simultaneously display laboratory prototype as a graph or planimetric view of imaged site for giving context while viewing volumetric data,
 - sheer computational requirements of 3-D visualization.

Another problem concerns the present inability to incorporate other more shape-oriented norms for determining the similarity of two spectral curves.

8. RECOMMENDATIONS FOR ENHANCED VIEWING CAPABILITIES

The stated goal is to be able to effectively view as a whole the high-dimensional data sets produced from imaging spectrometers, thereby providing some relief from the burden inherent in such a mass of data.

However, attempts at reduction in complexity should not result in such a severe loss of data as occurs with RGB maps of final classifications. Maps are fine for illustrating conclusions provided by automated methods, but do not readily provide enough information for examining intermediate results and making comparisons of conclusions with the original data.

Clearly volumetric rendering has potential in an expanded visualization role

to complement automated classification methods. However, in view of some of the concerns expressed in section 7, the applicability of this technique can be considerably enhanced by the inclusion of additional features and capabilities.

Some typical user scenarios which may be useful would involve the interplay between volumes of residuals, either in stereo or monoscopic projections, and cueing interesting regions, followed by more detailed examination of smaller volumes or planar sections. The ability to perform rotations of the volume would be especially crucial, as well as the ability to selectively "peel away" or "toggle off" obscuring regions to get a clearer view inside the region.

One of the useful tools for examining 3-D data would be a 3-D cursor with which to explore the volume and to specify planar sections or volume boundaries for removal or selection. The ability to use this 3-D cursor to insert and manipulate transparent "cutting planes" or raw image planes would be useful for generating a visual reference.

Additionally one could maintain a permanent plane for a color composite image at one end of the 3-D cube of residuals. Another possibility is to insert a projected view of the topography of the region on top of the 3-D cube to provide geometric context.

The user will want to make selections among types of display, such as volumetric, planar, single-pixel, or combinations. Delineations of planimetric regions and selections for laboratory prototypes will also be important. Concurrent displays of spectral prototype curves will be helpful for interpreting volumetrically rendered iso-surface shapes.

Finally, the user must have convenient facilities for selecting data types, data conditioning modes, and data exploration modes. A prototype sample screen which illustrates some of these options appears in Fig. 5. A processing history of options used during exploration sessions will allow the researcher to concentrate on visual information without having to worry about recording all the details of his functional selections.

Other scenarios which would involve change detection or interaction with the results of automated clustering applications would be:

- o rendering combined with automated clustering,
 - starter for automated methods,
 - perturbation of output from automated methods,
 - clustering domain,
 - full-pixel,
 - mixed pixel,
 - type of automated methods,
 - procedures,
 - expert systems,
 - examination of intermediate results of automated methods,
- o juxtapose 3-D perspective view of terrain onto image cube of spectral values,

- o insertion of quantitative labels and scales,
 - wavelength,
 - waveband number,
 - residual value,
 - confidence levels for calculated values.

An implementation using animation could cycle through various thresholds for displaying iso-surfaces of residual values. The corresponding dynamic changes in shapes of the intersections of these iso-surfaces with particular image planes may be helpful in determining appropriate residual thresholds.

Finally, more exploration of shape recognition measures may be required for effectively presenting the residuals between laboratory and actual spectral values.

9. REFERENCES

- o Adams, J.B., and Smith, M.O. (1986) "Spectral mixture modeling: a new approach to analysis of rock and soil types at the Viking lander site": Journal of Geophysical Research, 91, 8098-8112.
- o Boardman, J., (1990) "Inversion of Imaging Spectrometry Data Using Singular Value Decomposition", Proceedings IGARSS '90, College Park, MD, May 1990.
- o Bryant, J., "AMOEBRA Clustering Revisited", PE & RS, vol. 56, #1, Jan. 1990.
- o Dozier, J. (1988), "HIRIS-NASA's High Resolution Imaging Spectrometer for the Earth Observing System," IEEE Geoscience and Remote Sensing Newsletter, Nov. 1988.
- o Goetz, A., and Herring, M. (1989), "The High Resolution Imaging Spectrometer (HIRIS) for EOS", IEEE Trans. Geo. Remote Sensing, vol.27, #2, 1989.
- o Kruse, F.A., (1990a), "Analysis of Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) data for the northern Death Valley region, California and Nevada": In Proceedings of the 2nd Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) workshop, 4-5 June 1990, JPL Publication.
- o Kruse, F.A., (1990b), "Artificial Intelligence for Analysis of Imaging Spectrometer Data": Proceedings, ISPRS Commission VII, Working Group 2: "Analysis of High Spectral Resolution Imaging Data", Victoria, B.C., Canada, 17-21 September, 1990.
- o Levoy, M. (1988), "Display of Surfaces from Volume Data", IEEE Comp. Graphics & Applications, May 1988.
- o Levoy, M. (1990), "A Hybrid Ray Tracer for Rendering Polygon & Volume Data", IEEE Comp. Graphics & Applications, March 1990.

- o Mazer, A.S., Martin, Miki, Lee, Meemong, and Solomon, J.E., (1988), "Image processing software for imaging spectrometry data analysis": Remote Sensing of Environment, v.24, no.1, p.201-210.
- o Mustard, J.F., and Pieters, C.M. (1987a) "Abundance and distribution of ultramafic microbreccia in Moses Rock dike: Quantitative application of mapping spectroscopy": Journal of Geophysical Research, 92, B10, 10376-10390.
- o Mustard, J.F., and Pieters, C.M. (1987b) "Quantitative abundance estimates from bidirectional reflectance measurements": Journal of Geophysical Research, 92, B4, E617-E626.
- o NASA, (1987), "HIRIS, High-resolution imaging spectrometer: Science opportunities for the 1990s": Earth Observing System, Instrument panel report, V. IIC, National Aeronautics and Space Administration, 74 pp.
- o NASA (1986) "MODIS, Moderate-resolution imaging spectrometer": Earth Observing System, Instrument panel report, V. IIB, National Aeronautics and Space Administration, 59 pp.
- o Roberts, D.A., Yamaguchi, Y., and Lyon, R.J.P., (1985), "Calibration of Airborne Imaging Spectrometer Data to percent reflectance using field spectral measurements": in Proceedings, Nineteenth International Symposium on Remote Sensing of Environment, Ann Arbor, Michigan, October 21-25, 1985.
- o Rose, J.(1989), "Spatial Interference in the AVIRIS Imaging Spectrometer", PE & RS, vol. 55, #9, 1989.
- o Sabella, (1988)"A Rendering Algorithm for Visualizing 3D Scalar Fields", Proceedings SIGGRAPH, vol. 22, #4, Aug. 1988.
- o Salamonson, V. (1988), "The Moderate Resolution Imaging Spectrometer (MODIS)", IEEE Geoscience and Remote Sensing Newsletter, Aug. 1988.
- o Singer, R.B., and McCord, T.B. (1979) "Mars: Large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance": in Proceedings Lunar and Planetary Science Conference, 10th, pp. 1835-1848.
- o Smith, M.O., and Adams, J.B. (1985) "Interpretation of AIS images of Cuprite, Nevada using constraints of spectral mixtures": in Proceedings, AIS workshop, 8-10 April, 1985, JPL Publication 85-41, Jet Propulsion Laboratory, Pasadena, California, pp. 62-67.
- o Smith, M.O., Johnson, P.E., and Adams, J.B. (1985) "Quantitative determination of mineral types and abundances from reflectance spectra using principal components analysis": Journal of Geophysical Research, 90, C797-804.
- o Upson, C., and Keeler, M. (1988), "VBUFFER: Visible Volume Rendering", Proceedings SIGGRAPH, vol. 22, #4, Aug. 1988.

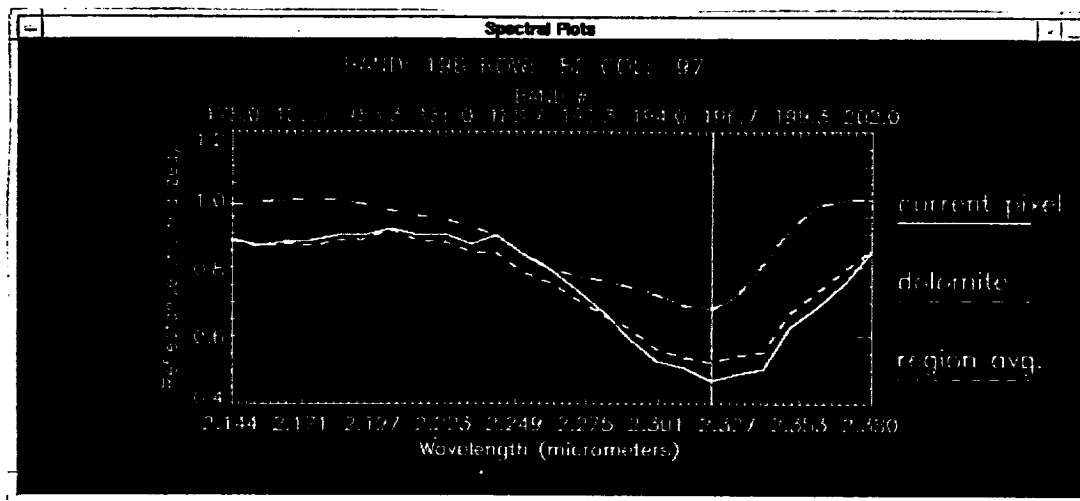


Fig. 1 (a). Dolomite Laboratory Spectrum and Sampled AVIRIS Spectra.

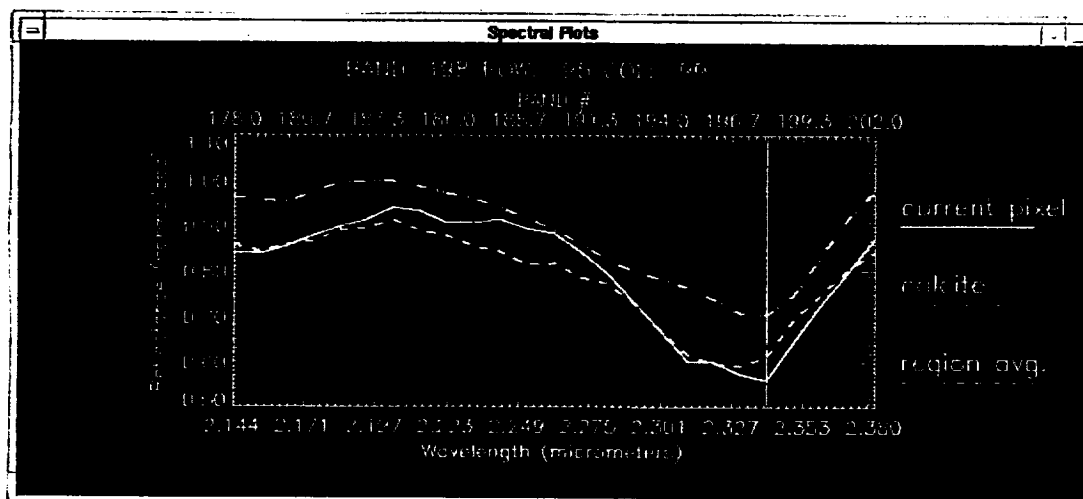


Fig. 1 (b). Calcite Laboratory Spectrum and Sampled AVIRIS Spectra.

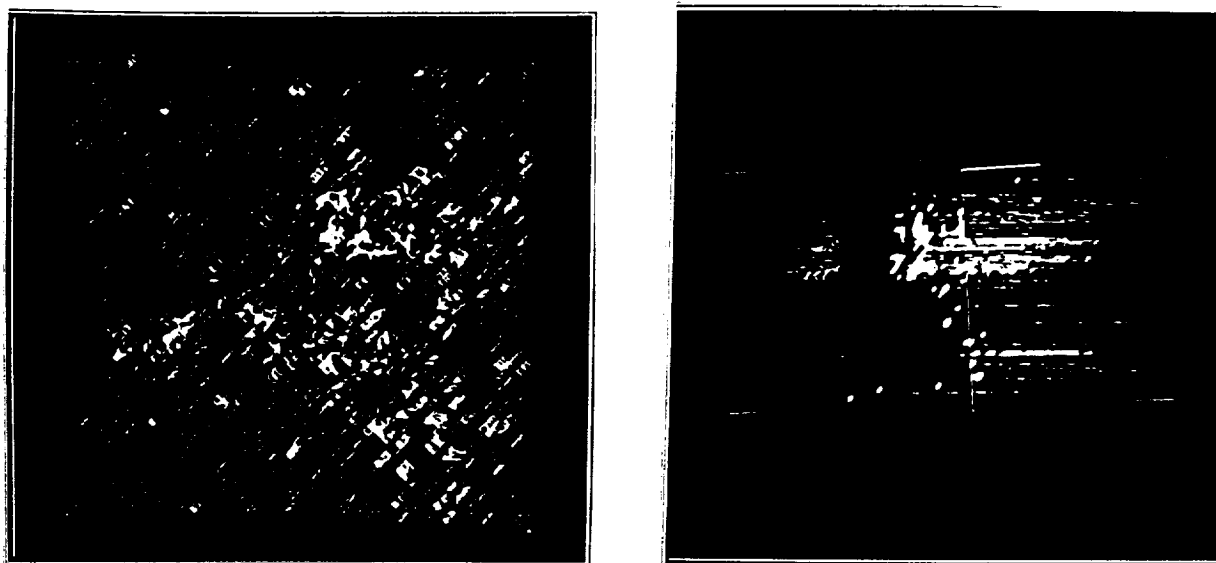


Fig. 2. Iso-Surface Renderings of Dolomite Residuals at Various Viewing Angles

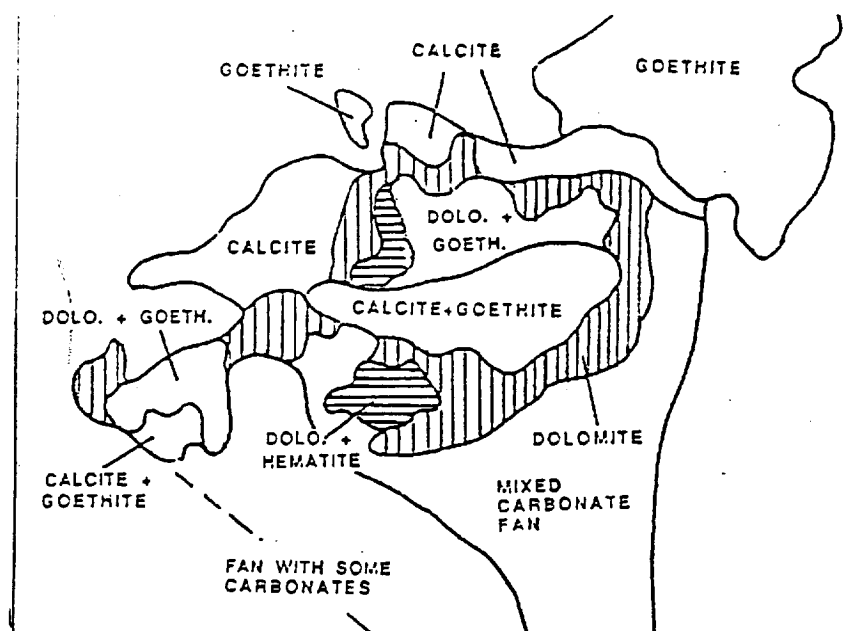
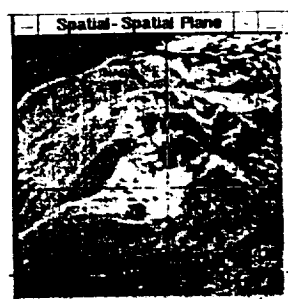
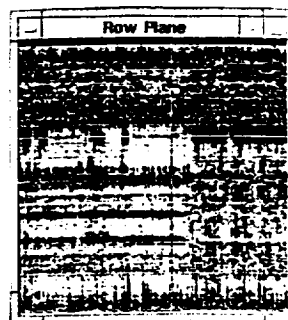


Fig. 3. Classification Map for Carbonate Area of Grapevine Mountain Region



(a)



(b)

Fig. 4(a) Raw AVIRIS Image Band #52, 4(b) Spectra for AVIRIS Line #82

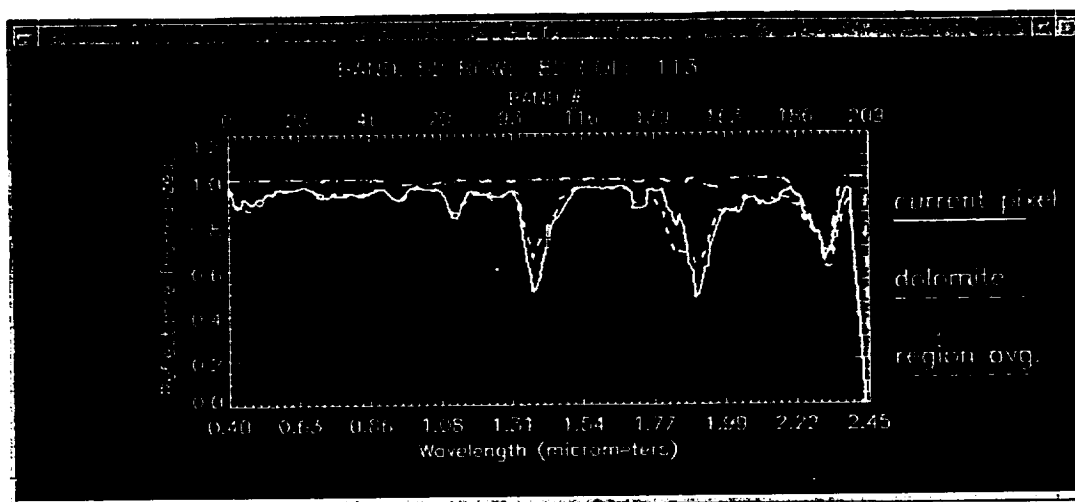
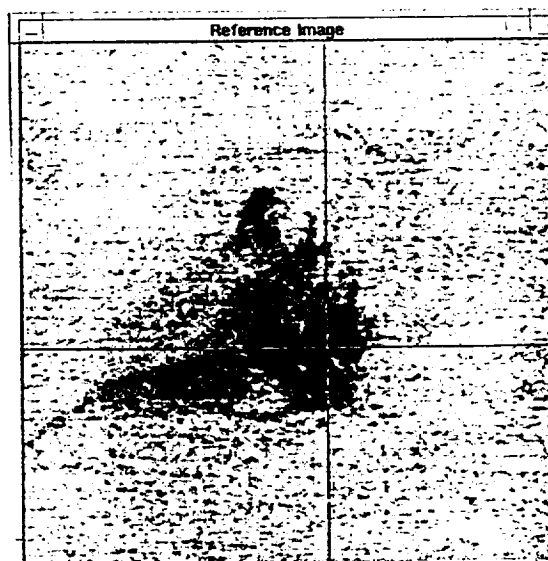


Fig. 4(c) Dolomite Laboratory Spectrum and Sampled AVIRIS Spectra

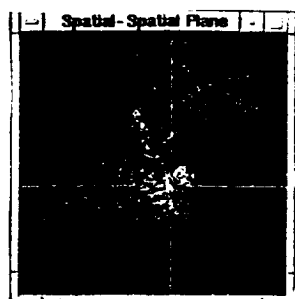


(d)

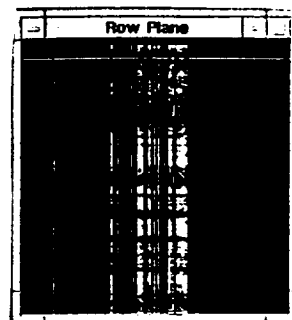


(e)

Fig. 4.(d) AMOEBA Cluster Map [Bryant,88] of AVIRIS Bands #189-203, (e) 2nd Principal Components of AVIRIS Clay-Mineral Bands



(f)



(g)

Fig. 4.(f) Planimetric Band #192 of Dolomite Residual Cube, (g) Line #82 Residual Spectrum for Dolomite

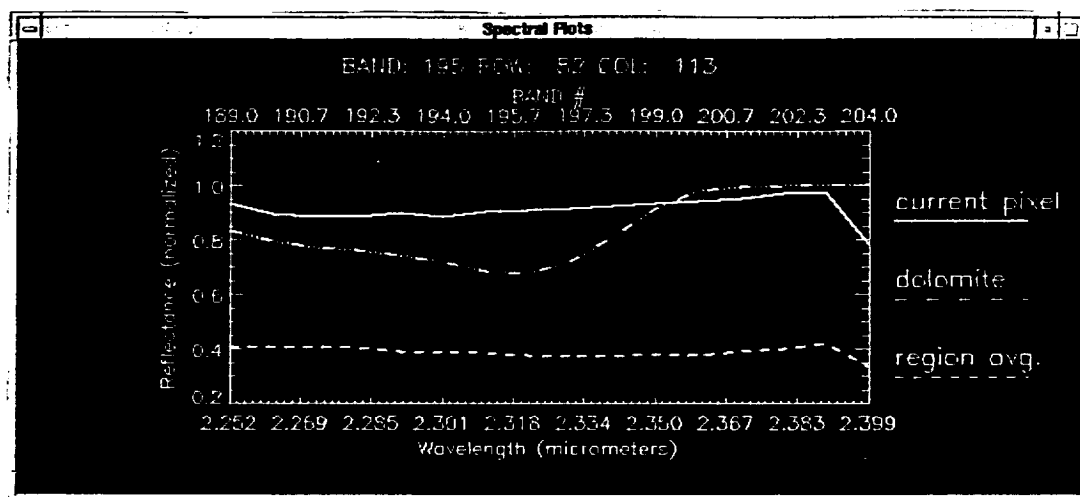


Fig. 4. (h) Dolomite Laboratory Spectra and Sampled AVIRIS Dolomite Residuals

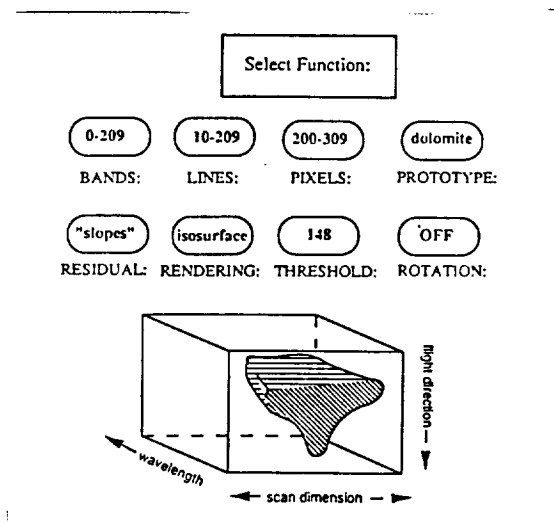


Fig. 5. Prototype Sample Screen

GENERATION, VISUALIZATION AND ANALYSIS OF IMAGE CUBES

M. Millot, K. Maurice, M. Karspeck, W. Kober and F. Leberl

VEXCEL Corporation
Boulder, Colorado, USA

ABSTRACT

The continuous increase in volume and diversity of remote sensing and collateral data implies that new methods of visualization and analysis must be devised to meet the challenge of efficiently processing these data. We discuss our solutions in the context of the future EOS era, applied to the two main classes of image cubes:

- Homogeneous image cube acquired by one instrument with multispectral capabilities, for example:
 - . Thematic Mapper,
 - . AVIRIS,
 - . Multispectral/Multipolarization SAR.
- Heterogeneous image cube composed of coregistered images coming from different sources, for example:
 - . Satellite image,
 - . Aerial photography,
 - . Digital Elevation Models and maps.

INTRODUCTION

The continuous increase in volume and diversity of remote sensing and collateral data implies that new methods of visualization and analysis must be devised to meet the challenge of efficiently processing the data. To generate image cubes containing different images acquired using different sensors with different geometric characteristics, an excellent coregistration is required. Once the image cube is generated, it can be visualized using several techniques. We investigated stereo visualization, animation and manipulation of the image cube as a 3D object, and insertion of a small window containing one layer into a different layer of the image cube. In addition to this visualization, the image cube may be analyzed. Among the numerous possible analysis to perform on an image cube, we implemented spectral analysis allowing the user to compare a pixel signature to a database of laboratory spectral prototype signatures.

IMAGE CUBE GENERATION

It is possible to categorize the various methods for producing coregistered images by the model used for the image deformation:

- . Empirical model: A general estimation of the mapping, between two images or between one image and the ground, is built without considerations regarding the sensor or the platform; this is sometimes referred as "warping" or "rubber sheeting" [1]-[2].
- . Physical model: Some data representing the geometric effects of the sensor and of the platform are available, so that a model of the physical phenomena can be found [3]-[6].

Independently of the model chosen, the coregistration of two images may be performed following two different processing paths. The first path begins with the acquisition, by manual or automatic (correlation or

matching) means, of common points between the images, and assumes that one image is the reference, while the other image is the image to resample. The other path assumes that there exists a third reference system (i.e the ground) and each image is registered to this third reference system. The images' coregistration then becomes an indirect result of this processing. The second processing path is the preferred way when image distortion due to relief is significant.

We experimented with the physical modeling on two types of images, SPOT (acquired from Spot Image Corporation) and NHAP (acquired from USGS) over Boulder, Colorado. The two multispectral SPOT images used had a viewing angle difference of 29 degrees, their mapping with the ground was estimated, using 1:24000 USGS maps, by the Kratky method [6]. Also, the two infrared NHAP images were scanned at a pixel resolution of 2.5 meters and a photogrammetric modeling was performed. Each image was resampled using the previously estimated mapping from the ground to the image and USGS 7.5 minute DEMs. The coregistration of the four images that resulted from this processing may be qualitatively checked using the visualization tools described later. Over Boulder, we also acquired a geocoded SEASAT image from JPL (Jet Propulsion Laboratory) and a geocoded THEMATIC MAPPER image from EOSAT.

An image cube may also contain GIS information. As an example, we included in our Boulder image cube rasterized hydrography 1:100,000 USGS DLG (Digital Line Graph), and contour lines computed from USGS 7.5 minute DEM's.

VISUALIZATION

We investigated three types of image cube visualizations, each one showing a different facet of a given image cube.

Stereo visualization

A software package called the Light Table has been developed by Vexcel. Currently, it provides basic functions for displaying stereo image pairs and measuring the parallax between features on those images. Two types of features can be created: single point and polyline. Features can be selected, deleted and selectively displayed or blanked. The Light Table was used to acquire the ground control points needed for the modeling of the SPOT and NHAP stereo pairs, and also to visualize what we call a "virtual image cube" (A virtual image cube is an image cube where the mapping between layers is known but applied only partially). As an example, after estimation of the mapping between the ground and the SPOT images, the DLG data and the contour lines were projected into the geometrically raw images, therefore allowing their visualization in stereo and adding the third dimension to data classically represented in two dimensions.

3D representation

A program provided with the Stardent 3000 computer for C/T and NMR volumetric visualization was the perfect prototype, possessing the basic functions we were envisioning. The display functions include panning, zooming (in predefined increments), brightness/contrast adjustment, overall image cube display, layer selection, arbitrary slicing through the image cube, animation with adjustable speed and graphical display of the line and column intensities defined by the cursor position. It allows the user to qualitatively detect spectral changes (for example using an AVIRIS image), to display the spectral response of a given line or column, and using the animation to check and use the layers coregistration.

Region Of Interest (ROI)

Using the X Window System and low level Stardent routines, we were able to implement two instances of the ROI:

- . Zooming: using a virtual image cube containing two images coregistered but at two different resolu-

tions (called image low and image high in this paragraph), display image low as the background and inside a smaller window whose center is defined by the cursor position, display:

- . image high at the resolution of image low,
- . image high at an intermediary resolution,
- . image high at full resolution,
- . continuously interpolated image low from its original resolution to the image high resolution,
- . continuously interpolated image high from the image low resolution to its original resolution,
- . continuously a weighted average of the two interpolated images at different resolutions. It can be represented mathematically as :

$$(1-\beta) \cdot IL + \beta \cdot IH$$

β is a function of the zooming factor, IL and IH are the grey levels of the low and high resolution images respectively.

. Multiple layer display: Selecting one layer as the background image, display inside it a smaller window containing other layers or any arithmetic combination of several layers. The position of the small window inside the background image is controlled dynamically by a mouse.

ANALYSIS

Using PV WAVE (from Precision Visuals Inc), we built the spectral analysis program [7]. Five windows are available to the user:

- . A window (called window 1 here) containing one fixed layer of the homogeneous image cube. Using the cursor, the user indicates a pixel in the image or defines an area by drawing a closed polygon.
- . A window (called window 2 here) containing the spectral response of the entire line indicated by the cursor in window 1. Using the cursor inside this window, the user indicates the layer she/he wishes to be displayed.
- . A window containing the spectral band indicated by the cursor in window 2.
- . A window (called window 4 here) containing a graphical representation of the spectral response of the area or pixel defined in window 1. Inside this window, the user can sequentially display the response of laboratory prototypes and visually compare the image response with laboratory prototype responses.
- . A window containing a scale used to define the spectral interval used as the X axis in window 4. The user can adjust this in order to focus on a particular area of the spectral dimension.

This program was tested using a calibrated AVIRIS image of Death Valley and some normalized mineral spectral responses.

CONCLUSION

We investigated the three important parts of the image cube processing: generation, visualization, and analysis. Although a lot remains to be done to finalize and integrate this work, we strongly believe that the image cube concept associated with powerful computers and window systems, will be a good way to present and analyze data during the EOS era.

ACKNOWLEDGMENTS

This paper presents one portion of the research completed under SBIR phase II contract number NAS7-1070, sponsored by the National Aeronautics and Space Administration.

REFERENCES

- [1] G.E Ford and C.I.Zanelli "Analysis and quantification of errors in the geometric correction of satellite images" *Photogram. Eng. Remote Sensing*, vol. 51, No. 11, pp. 1725-1734, 1985.
- [2] A. Goshtasby " Registration of images with geometric distortions" *IEEE Trans. Geosci. Remote Sensing*, vol. 26, no. 1, pp.60-64, 1988.

- [3] M. Naraghi, W. Stromberg, M. Daily " Geometric rectification of radar imagery using digital elevation models" *Photogram. Eng. Remote Sensing*, vol. 49, no. 2, pp. 195-199, 1983.
- [4] R. Kwok, J.C. Curlander, S.S Pang " Rectification of terrain Induced distortions in radar imagery" *Photogram. Eng. Remote Sensing*, vol. 53, no. 5, pp. 507-513, 1987.
- [5] T. Westin " Precision Rectification of SPOT imagery" *Photogram. Eng. Remote Sensing*, vol. 56, no 2, pp. 247-253, 1990.
- [6] V. Kratky " Rigorous photogrammetric processing of SPOT images at CCM Canada" *ISPRS J. of Photogram. and Remote Sensing*, vol. 44, pp. 53-71, 1989.
- [7] A.S. Mazer, M. Martin, M. Lee and J.E. Solomon " Image processing software for imaging spectrometry data analysis" *Remote Sensing Environ.*, vol. 24, pp. 201-210, 1988.

Concepts for Processing and Analyzing of Multiple SAR and Landsat Images

J. Thomas, R. Mullen, F. Leberl, W. Kober
VEXCEL Corp.
Boulder CO.

J. Cimino
Jet Propulsion Laboratory
Pasadena CA.

ABSTRACT

The use and application of multi-sensor data sets are an absolute necessity for remote sensing to fulfill its promise. Initial efforts of registering dissimilar images have been made, but results are not widely available. This paper describes some initial work to better understand the difficulties of multi-sensor registration. Accuracies achieved are in excess of a pixel diameter.

Key Words: Multi-sensor, registration, SAR, Landsat, SIR-B, registration accuracy.

1. Introduction

The present paper reports on some preliminary results of an empirical study to examine the problems of interactive registration of differing selected imagery types. This problem domain is complex due to dissimilarities in images that need to be matched, i.e. registered or fused.

Another purpose is to understand and interpret corresponding image intensity variations. In particular, those variations to be examined are functions of:

- wavelength;
- incidence angle;
- polarization;
- time of image acquisition;
 - illumination changes
 - scene content changes
- type of sensor

An eventual goal will be a capability for the creation of data products from multi-sensor, multi-temporal sources, which delineate regions corresponding to intensity variations as a function of selected variables.

Dissimilar image matching has an extensive literature. An early contribution by [Mader, 76] addressed SAR and Landsat-3-RBV. Many similar efforts followed, essentially relying on manual techniques. A recent review by Leberl (1986) enumerated 90 references. Automating the task reliably remains elusive. Even matching ascending and

descending orbit SAR images can pose great difficulties because of local intensity reversals. In [McConnell, 87] the binarized regions derived from Marr-Hildreth edges are employed to match orbit in variant SAR image contents. Edges and invariant moments were used in [Wong, 79] for SAR-optical imagery matching.

The following will describe a more elaborate dataset than is commonly used, but it is still short of what is expected in the EOS-era [EOS, 88]. The following discussion describes some of the difficulties involved in achieving sub-pixel registration accuracy.

2. Description of Experimental Datasets

The following experimental dataset was used to develop experience in multi-sensor processing.

The sensor source and quantities of each image type were:

- Landsat TM (7 bands);
- SIR-B SAR (1 image);
- Aircraft SAR (2 overflights)

Relevant available parameters for each type are described in Table 1.

The imaged scene was Raisin City, CA. The topography is flat, and the major man-made features are Raisin City itself and Henderson Road. The foliage types consist of orchards and various crops. The orchards contain walnuts, peach, plum and almond. The crops were alfalfa, corn, beans, tomatoes, and cotton. The relevant crop descriptors are:

- % ground cover by crop type;
- crop state;
- time period;
- location

3. Registration Methodology

Our experimental work is designed to provide experience that will lead to the development of automated routines for multi-sensor image registration.

Achieving the highest fidelity co-registration of all imagery sets to a map, i.e. geocoding, is expected to require first registering the highest resolution imagery, in this case the aircraft SAR, to map coordinates. The subsequent registration of lower resolution imagery to this higher resolution data set can be accomplished either by up-sampling the former or down-sampling the latter.

If centralized terrain topography were involved in the imaged scene, then added procedural complexity would be present. For SAR images, geocoding methods for real satellite imagery [Kwok, 87], [Curlander, 82,84,87] could be used with modification for real aircraft SAR images. Such modifications would be required to take into account the differing phase compensations for the aircraft scenario from the satellite case. But because the imaged scene is topographically flat, such terrain compensations were not required in this case.

However, another problem arises with SAR because the aircraft slant plane SAR data

is complex quadpole polarization imagery. This allows any desired combination of transmit and received polarizations to be incorporated into any real slant plane SAR image. Therefore, the most efficient method would be to first geocode the complex data. Then any real SAR image with desired polarization combinations subsequently created from this geocoded complex data would automatically be geocoded.

However, geocoding complex data requires interpolating phase as well as amplitude data. Efficient methods for interpolating the more sensitive phase data are the subject of ongoing research. No results are presented using complex interpolation.

Therefore, each real slant plane aircraft SAR image, for a particular polarization combination, was first created and then separately geocoded. The first aircraft SAR image was registered to a 1:24,000 map of Raisin City to yield the base image. The control points that were used were road intersections and other well defined points in both the image and map.

Examples of the SIR-B SAR, aircraft SAR, and Landsat images are shown in Fig. 1-3 respectively. The mapping of the control points leads to a so-called deformation grid, shown in Fig. 4. This grid is then used to generate the final resampling of pixels for a geocoded image. The resampled aircraft SAR image is shown in Fig. 5. The effects of range curvature are clearly evident.

The actual registration quality depends on an ability to use context in the matching process. Automated techniques incorporate such context by matching areas and edges. Manual techniques need an ability to interactively shift the two datasets over one another.

The other images were also registered to the map using similar control points. There were some clearly visible

misregistrations relative to the base image, as shown in Fig. 6-7.

To reduce these registration errors, the remaining images were re-warped again to fit the base image by the use of corresponding control points in the images. These results are shown in Fig. 8-9.

The number of control points used, their spatial distribution, and the resulting deformation grids were plotted in each case. Using the deformation grid, the errors at each of the control points were calculated, as shown in Fig. 4. A summary of the registration error statistics is tabulated in Table 2.

4. Preliminary Discussion of Results

It is to be noted that the overall accuracies of multi-sensor registration varies in excess of one pixel for all image-to-map registrations. Image-to-image registration also tended to have residual errors of several pixels at the higher resolution, but one pixel or less at the lower resolution.

The slant range format of the aircraft SAR images involves considerable distortion from a ground plane geocoded image. In addition, the noise and speckle present in both the aircraft and SIR-B imagery made the identification of control points difficult.

The identification of common control points in the dissimilar image sets with the corresponding control points in the base image was also difficult because of intensity variations arising from differences in wavelength. The problem of relating TM data to radar was especially severe. This is not surprising given earlier reports about similar efforts [Heberl, 86].

The problem of recognizing common regions with intensity differences dependent on wavelength effects is not tractably

wavelength effects is not tractably resolved using local histogram remapping. In general, this problem is overcome using common perceived edges. However, this process is difficult to automate because there are edges which are not common to all imagery types and edges tend to break up differently even among the radar images.

5. Conclusions

Interpretation of intensity differences in multi-sensor images is an ongoing activity. A need exists to develop automated techniques for matching of dissimilar images. Subsequent to that, a qualitative understanding needs to be developed relating image intensities to physical properties of imaged objects, such as crop descriptors, and to sensor parameters.

Considerable effort has been expended in past studies of multi-sensor datasets. Unfortunately, the sheer difficulty of the initial data registration task usually resulted in a scenario where most of a project's time and resources were spent on registration. Consequently, the actual study of registered image contents and their parametric variations tended to suffer.

It is clear that improved technology for registration must be developed to reduce the drudgery and wasted effort of mundane data preparation from the scientific analysis of multi-sensor datasets.

We have begun to study avenues for automating the multi-sensor registration process and report on first qualitative results. We conclude that sub-pixel accuracies will be a difficult goal to reach.

References

1. Kwack, R. et al., "Rectification of Terrain Induced Distortions in Radar Imagery," Photog. Eng. Rem. Sens., vol. 53, #5, May 1987.
2. Curlander, J., "Location of Spaceborne SAR Imagery," IEEE Trans. Geo. Rem. Sens., GE-20, #3, July 1982.
3. Curlander J., "Utilization of Spaceborne SAR Data for Mapping," IEEE Trans. Geo. Rem. Sens., GE-22, #2, Mar. 1984.
4. Curlander J. et al., "A post-processing system for automated rectification and registration of spaceborne SAR imagery," Int. J. Rem. Sens., vol-8, #4, 1987.
5. Mader, G., McCaudless W., "Space Based Topographic Mapping Experiment Using Seasat-SAR and Landsat-3-RBV Imagery", Phoenix Corp., McLean, VA 22102, 1981.
6. Leberl, F., "Review of Basic Techniques for Multi-Sensor and Image Time Series Analysis", VEXCEL Technical Report #100986 (to European Space Agency), Boulder, CO 80201, Oct. 1986.
7. McConnell, R., "Matching of Dissimilar Radar Images Using Marr-Hildreth Zero Crossings", Proc. Amer. Soc. Photog. Rem. Sens., Balt., MD, April 1987.
8. From Pattern to Process: The Strategy of the Earth Observing System, EOS Science Steering Committee Report Vol. 11 NASA, 1988.
9. Wong, R., "Radar to Optical Scene Matching", Proc. SPIE, Vol. 186, 1979.

A) Aircraft SAR
 Resolution = 7.495 m in range; 10.98 m in azimuth
 Image size = 1024 x 1024 pixels
 Wavelength = L-Band
 Polarization = complex quadpole
 Flying height = 12221 m
 Incidence angle
 Time of acquisition
 - 9/28/84 21:04:55
 - 9/28/84 20:50:38

B) SIR-B SAR
 Resolution = 34.6 m ground range; 28.5 m azimuth;
 12.5 m pixel size
 Image size = 7000 x 2957
 Wavelength = L-Band (23 cm)
 Polarization = HH
 Flying height = 235 km
 Incidence angle = 24.0 deg.
 Time of acquisition
 - 284/ 12:17:26.744 (1984)
 Ascending orbit

C) Landsat TM
 Resolution = 30 m Bands 1-6; 120 m Band 7
 Image size = 4320 x 2984
 Wavelength
 Band 1 = .45 - .52 um
 Band 2 = .62 - .68 um
 Band 3 = .63 - .69 um
 Band 4 = .76 - .90 um
 Band 5 = 1.55 - 1.75 um
 Band 6 = 10.40 - 12.50 um
 Band 7 = 2.08 - 2.35 um
 Flying height = 705 km
 Time of acquisition = not available

Parameters Of The Multi-Sensor Dataset

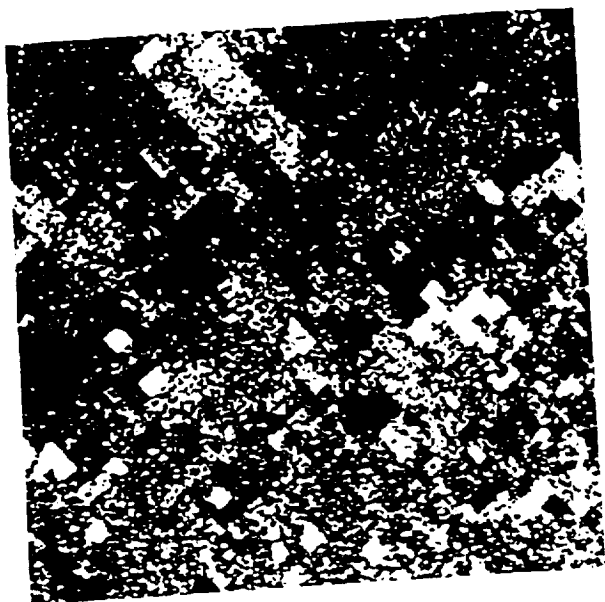
Table 1

Image Types Matched	RMS Pixel Errors		Pixel Size	RMS Metric Errors		Total # Pixels	# Controls
	X	Y		X	Y		
1) Aircraft #1 to Map	2.83	1.40	10.0m	28.3m	14.0m	1200 x 1100	25
(2) SIR-B to Map	1.66	1.95	12.5m	23.3m	24.4m	1200 x 1200	30
(3) (2) to (1)	2.39	1.29	10.0m	23.9m	12.9m	1100 x 1000	37
(4) TM to Map	0.40	0.39	30.0m	12.0m	11.7m	1300 x 1300	30
(5) (4) to (1)	2.35	1.55	10.0m	23.5m	15.5m	1100 x 1100	38
(6) Aircraft #2 to Map	1.79	1.30	10.0m	17.9m	13.0m	1200 x 1100	21
(7) (6) to (1)	2.40	2.27	10.0m	24.0m	22.7m	1100 x 1000	54

Summary Statistics of Registration Errors

Table 2

ORIGINAL PAGE IS
 OF POOR QUALITY



Original SIR-B SAR Image

Figure 1



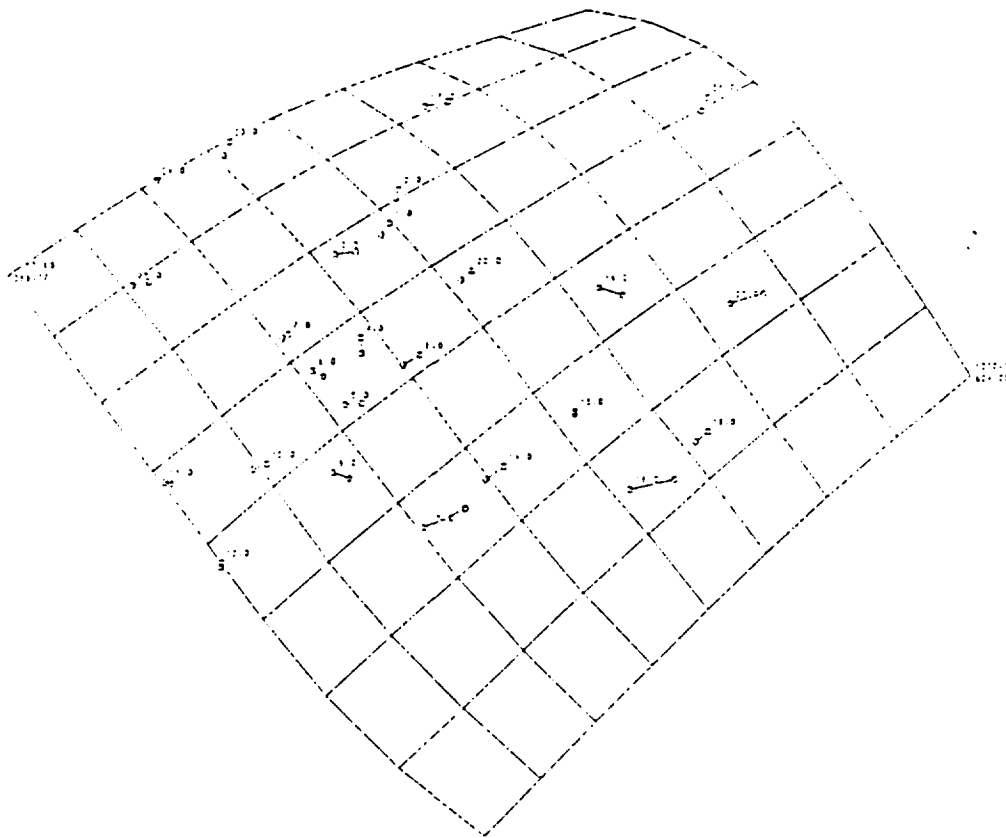
Original Landsat TM Image
Outer Box Registered to Map
Inner Box Registered to Map Registered Aircraft SAR
Figure 3



Original Aircraft SAR Image

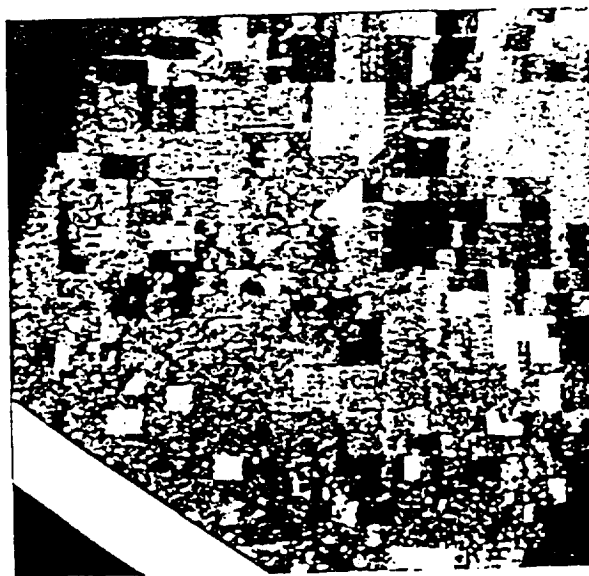
Figure 2

ORIGINAL PAGE IS
OF POOR QUALITY



Registration Deformation Grid
for Aircraft SAR Image

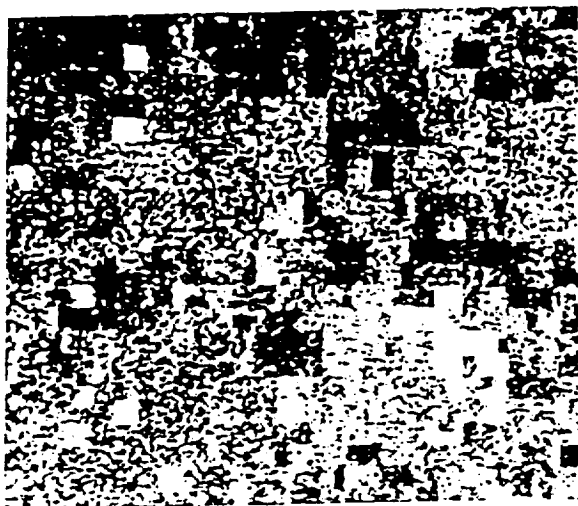
Figure 4



Aircraft SAR Image Registered to Map

Figure 5

ORIGINAL PAGE IS
OF POOR QUALITY



SIR-B Image Registered to Map

Figure 6



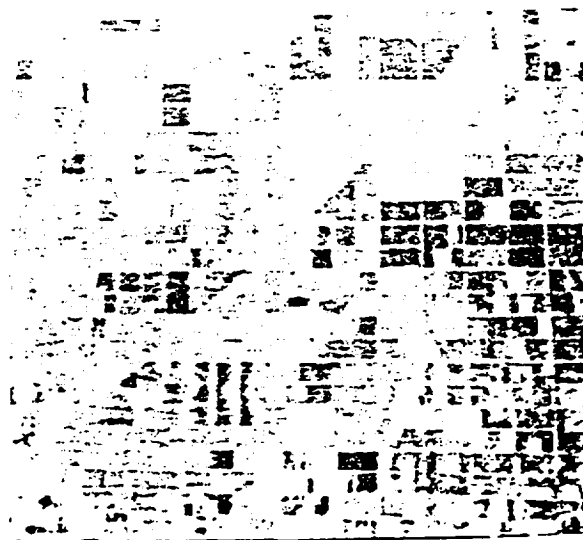
SIR-B Image Registered to Geocoded
Aircraft SAR Image

Figure 8



Landsat TM Image Registered to Map
TM Image Registered to Map
Box Corresponds to Inner Box of Fig. 3

Figure 7



Landsat TM Image Registered to
Geocoded Aircraft SAR Image

Figure 9

ATTACHMENT I

SPOTCHECK: SOFTWARE FOR GEOMETRIC PROCESSING OF SINGLE SPOT IMAGES AND OF STEREO PAIRS

OVERVIEW OF METHOD

1. Introduction

This document provides an overview of V. KRATKY method corresponding to my understanding after his visit from 07-17-90 to 07-19-90. The intent of this document is to give some information for the modification and integration in the EOS workstation. It focuses on method understanding and not in software information which is described in the SPOTCHECK Manual.

2. Least Mean Square

Before going to the method description, this paragraph is a brief summary of the Least Mean Square Method (extracted from the Manual of Photogrammetry of Basic Mathematics of Photogrammetry). The problem model is the following:

n unknowns (X_i)

r measured parameters (Y_i)

m independent equations $f_j((X_i)(Y_i)) = 0$

the parameters can be represented by

$$Y_i = Y_i^{oo} + v_i$$

with Y_i^{oo} the measured value and Y_i the ideal value also with (X_i^o) some approximation of the unknowns. Each of the m f_j functions can be approximated by:

$$f_j((X_i),(Y_i)) = f_j((X_i^o),(Y_i^{oo})) + \left(\frac{\partial f_j}{\partial Y_1}\right)^o v_1 + \dots + \left(\frac{\partial f_j}{\partial Y_r}\right)^o v_r + \left(\frac{\partial f_j}{\partial X_1}\right)^o \Delta X_1 + \dots + \left(\frac{\partial f_j}{\partial X_r}\right)^o \Delta X_r$$

which in matrix form leads to:

$$AV + Bg + u = 0$$

$$\begin{aligned} u &= f_i^o \\ V &= (v_i) \\ g &= (\Delta X_i) \end{aligned}$$

The purpose of the Least Mean Square Method is to minimize $\sum p_i v_i^2$ with p_i representing the weight associated with each measurement. After some derivations, the general normal equations are:

$$B^T (A P A^T)^{-1} B g = -B^T (A D A^T)^{-1} u$$

which represents the general formulation of Least Mean Square adjustment.

3. General Understanding of Kratky Method

3.1 Collinearity Condition

3.1.1 General Presentation

This description found in the chapter of *Online Non-Topographic Photogrammetry in Non-Topographic Photogrammetry* contains a description valid for the SPOT case, especially the normal equations and the way to handle the points unknowns.

The condition equations, called collinearity conditions, express the fact that the image point and the object point are collinear. His formula is a derivation of the classical collinearity equations more precisely. He is writing:

$$F_x = \Delta X_z' - \Delta Z_x' = 0 \quad F_y = \Delta Y_z' - \Delta Z_y' = 0$$

which represents 2 equations of the fact that the vectorial product is equal to zero

$$\begin{vmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{vmatrix} * \begin{vmatrix} x' \\ y' \\ z' \end{vmatrix} = \begin{vmatrix} \Delta X_y' - \Delta Y_x' \\ \Delta X_z' - \Delta Z_x' \\ \Delta Y_z' - \Delta Z_y' \end{vmatrix}$$

$$\text{if } \Delta X_z' - \Delta Z_x' = 0 \text{ and } \Delta Y_z' - \Delta Z_y' = 0 \text{ then} \\ \Delta X_y' - \Delta Y_x' = 0$$

using classical derivation he arrives to the following formula:

$$Av + Bg + u = 0 \text{ and } P \text{ weight matrix with}$$

$u:$	residual of the 2 equations
$g : [g_0 g_x]^T$	g_0 orientation parameter
	g_x points unknowns
$v : (v_x, v_y)$	error of measurements

which leads to the following normal system:

$$B^T P B g + B^T P u = 0, \text{ new } P = (A \hat{Q} A^T)^{-1}$$

after splitting B into $[B_0 B_x]$ one can write the following system of equations

$$B_0^T P B_0 g_0 + B_0^T P B_x g_x + B_0^T P u = 0$$

$$B_x^T P B_0 g_0 + B_x^T P B_x g_x + B_x^T P u = 0$$

using the second equation we can find the points unknown as

$g_x = -(B_x^T P B_x)^{-1} B_x^T P (B_0 g_0 + u)$ so yielding to a reduced system of normal equations

$$B_0^T P_0 B_0 g_0 + B_0^T P_0 u = 0 \text{ with } P_0 = P - P B_x (B_x^T P B_x)^{-1} B_x^T P$$

This system is solved using classical inversion of matrices. The interesting point is the following:

- if the point has known ground coordinates, therefore, the point coordinates become constants and then $B_x = 0$ (first derivative versus points coordinates) and the system simplifies to $P_0 = P$.
- if the points are known only by some ground coordinates, this zeros some of the lines and columns of the B_x matrix therefore making the $(B_x^T P B_x)$ matrix singular. At this point, a generalized numerical technique is used to still be able to invert this matrix, this leading to a new system of equations. This approach differs from the classical photogrammetric solution where in the case of known ground coordinates, some additional equations are used to consider the discrepancies between measured and approximated coordinates. The main advantages of doing this is the removal of the points unknowns. The main disadvantage seems that there is no possible considerations for the possible ground noise. A technique described in *NON-TOPOGRAPHIC PHOTOGRAMMETRY* simulate on the weight matrix the presence or absence of known ground coordinates is not used in the SPOTCHECK package.

3.1.2 SPOT Application

The main difference between aerial photography and SPOT is the time component of the image acquisition. The main assumption behind this software is that SPOT, during the image acquisition, stays on a Keplerian orbit defined by nominal values of the satellite position. The formalism is the same as for aerial photography. The collinearities functions are:

$$F_x \equiv \Delta X z - \Delta Z x = 0 \quad F_y \equiv \Delta Y z - \Delta Z y$$

with the time dependent parameters given by:

- Position of SPOT:

$X_i = X_o + y'\dot{X} + y'^2\ddot{X}$ with \dot{X} , \ddot{X} computed from approximated Keplerian parameters, and X_o unknown satellite position for the center of scene.

- attitude angles:

$\alpha = \alpha_o + y'\dot{\alpha} + y'^2\ddot{\alpha}$ with α_o , $\dot{\alpha}$, $\ddot{\alpha}$ unknowns to be estimated.

In addition to these unknowns, he introduces a focal length correction and a quadratic term for the pixel number ($x' = x' + Dx'^2$, D is the unknown).

Once the unknowns are defined, the method is identical to the method previously described (see Rigorous Photogrammetric Processing of SPOT images at CCM Canada).

3.1.3 Constraints

3.1.3.1 R Constraint

This constraint states that the distance between the Earth center and the satellite must be equal to the Earth radius plus the flying height:

$$F_c \equiv X_c^2 + Y_c^2 + Z_c^2 - (R + h)^2 = 0$$

considering that $X_c \equiv X_o + y'\dot{X} + y'^2\ddot{X}$ and the measurement is the radius of the orbit, the analytical formulation can be derived using the same technique described previously (see V. Kratky note dated 8-Jan-87).

3.1.3.2 Traveled Angle Constraint

This constraint states that the difference between the "measured" travel angle and the computed travel angle is equal to zero:

$$F_c \equiv \tau_p - \tau_o = 0$$

Assuming that the travel angle velocity is constant and equal to -0.12, the analytical formulation can be derived and is identical to the one previously described (see V. Kratky note dated 23-Oct-87).

3.1.3.3 Longitude of the Ascending (Descending) Node

This constraint states that the difference between the “measured” longitude at the computed longitude is equal to zero:

$$F_c \equiv \lambda_p - \lambda_o = 0$$

No analytical formulation are currently available.

ATTACHMENT J

AVS NETWORKS PRELIMINARY DESIGN DOCUMENT

NASA - Jet Propulsion Laboratory
SBIR Phase II Grant

**EOS PROTOTYPE IMAGE ANALYSIS
WORKSTATION**

AVS-NETWORKS PRELIMINARY DESIGN

**Prepared by
VEXCEL Corporation**

2 July 1990
Boulder, Colorado

This represents a preliminary document for internal use by the Project

TABLE OF CONTENTS

1.	INTRODUCTION	1
2.	AVS NETWORKS SAMPLES	1
	2.1 MODULES INPUTS AND OUTPUTS	2
3.	RESAMPLING	3
4.	GEOMETRIC PROCESSING OF FEATURES	5
5.	IMAGE CUBE GENERATION	11
6.	CONCLUSION	14

1. INTRODUCTION

The purpose of this document is to demonstrate the feasibility of implementing the EOS Workstation functions as described in the EOS Workstation document inside the Application Visualization Software (AVS) available from STARDENT. AVS allows the user to:

- Visualize geometric, image and volume datasets;
- Interactively construct and execute a processing network using a visual network editor;
- Connect user computational programs to the processing network;
- Build specific applications including user-written modules.

This represents attractive capabilities for EOS-type image analysis. In order to assess the feasibility of this approach, we designed AVS-networks for three image processing functions, namely resampling, geometric processing and image cube generation. AVS currently is available on STARDENT's line of computers and is also being transferred to other vendors' platforms, in particular to SUN.

2. AVS-NETWORKS

The AVS-network editor provides a visual programming interface which uses on-screen point and click operations to design applications as networks of modules. A "module" is a software element performing a specific user-relevant function. In AVS, modules can be dynamically added, connected and deleted. Modules are only re-executed when new data are required or an input parameter is changed.

Modules have control panels for interactive control of input parameters by on-screen sliders, file browsers, dials and buttons, or input devices such as dials (in particular STARDENT's knob box). The control panel is automatically generated when a module is connected into the network.

Using these tools, we have designed sample networks for resampling, geometric processing of features, and image cube generation. These were chosen because they are representative of the functions provided by the EOS Workstation; each function can be applied in different environments, which will demonstrate the power of the AVS approach over the traditional programming approach. These functions share common modules, therefore showing how modules are used in different applications.

2.1 Input and Output Modules

AVS offers a "flow executive". Each module is defined completely to this executive via inputs, outputs and parameters. The inputs and outputs of each module actually must be one of the following AVS data types:

- Byte,
- Integer,
- Floating point number,
- Text string,
- Colormap (lookup table),
- Field (generalization of array),
- Geometry (3D object),
- Pixel map (X pixmap).

The last three types are the ones mostly used by the existing AVS modules. Several problems arise when one tries to use them for the EOS Workstation modules:

- The only attributes associated with these data types are color and dimension. But for the EOS Workstation we would like to associate with each image some parameters describing the geographical location, date of acquisition, etc, and make some of this information available to modules or to the user.
- Actually AVS loads each of these object types into memory before it is used. If physical memory is unavailable, the system will store part of the object on disk as memory "pages". But this approach, with the associated paging, can reduce considerably the system's performance when the user is dealing with large images containing several spectral bands.

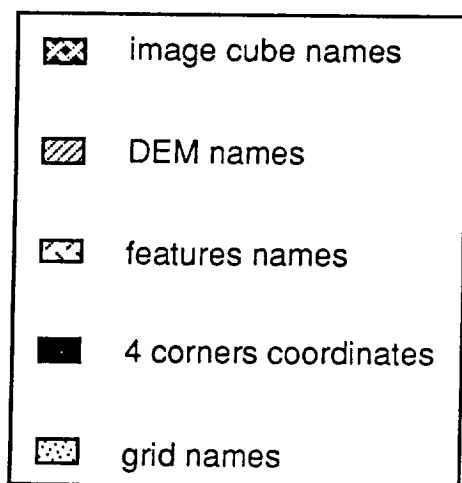


FIGURE 1
INPUT-OUTPUT DEFINITION

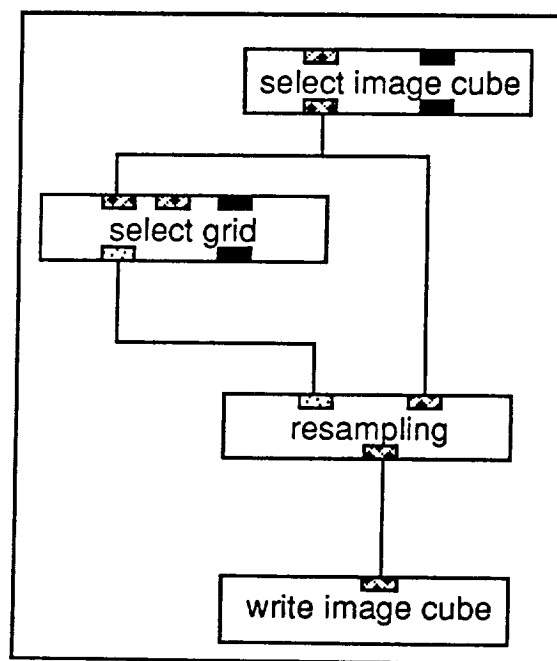


FIGURE 2
RESAMPLING (IMAGE AND GRID)

STARDENT is planning to enhance AVS by allowing the user to define his own data types and by modifying the object management. But to circumvent the current limitation, we decided to use an existing data type: the text string. It contains the object names or some parameters (see Figure 1 for the list of object names and their associated symbols). With this technique, the responsibility for accessing and creating each object is given to the modules rather than to the AVS flow executive.

3. RESAMPLING

Image resampling is an important part of the EOS Workstation, and can be used in many different contexts. In order to be able to comprehend the following examples, it is useful to be familiar with VEXCEL's proposed procedures for EOS-type image processing. We defined AVS networks for the following instances:

- (a) Resampling of an image using a grid of anchor points (see Figure 2). This requires the following modules:
 - Select image cube.
 - Select grid of anchor points, using the image cube name.
 - Resampling; at this stage most of the user options are exercised.
 - Write image cube to transform a temporary into a permanent file.
- (b) Resampling of an image using a grid of anchor points and a DEM (see Figure 3). This is composed of the following modules:
 - Select image cube.
 - Select grid.
 - Select DEM. The DEM is chosen using the grid's 4 corner coordinates.
 - Generate new grid. The existing grid is modified to account for the elevations provided by the DEM.
 - Write image cube.
- (c) Resampling of an image using the associated auxiliary data and a DEM (see Figure 4). This is applicable to SAR images, and it is composed of the following modules:
 - Select image cube.
 - Select DEM.
 - Physical resampling. The auxiliary data are indirectly referenced by the image cube name.
 - Write image cube.

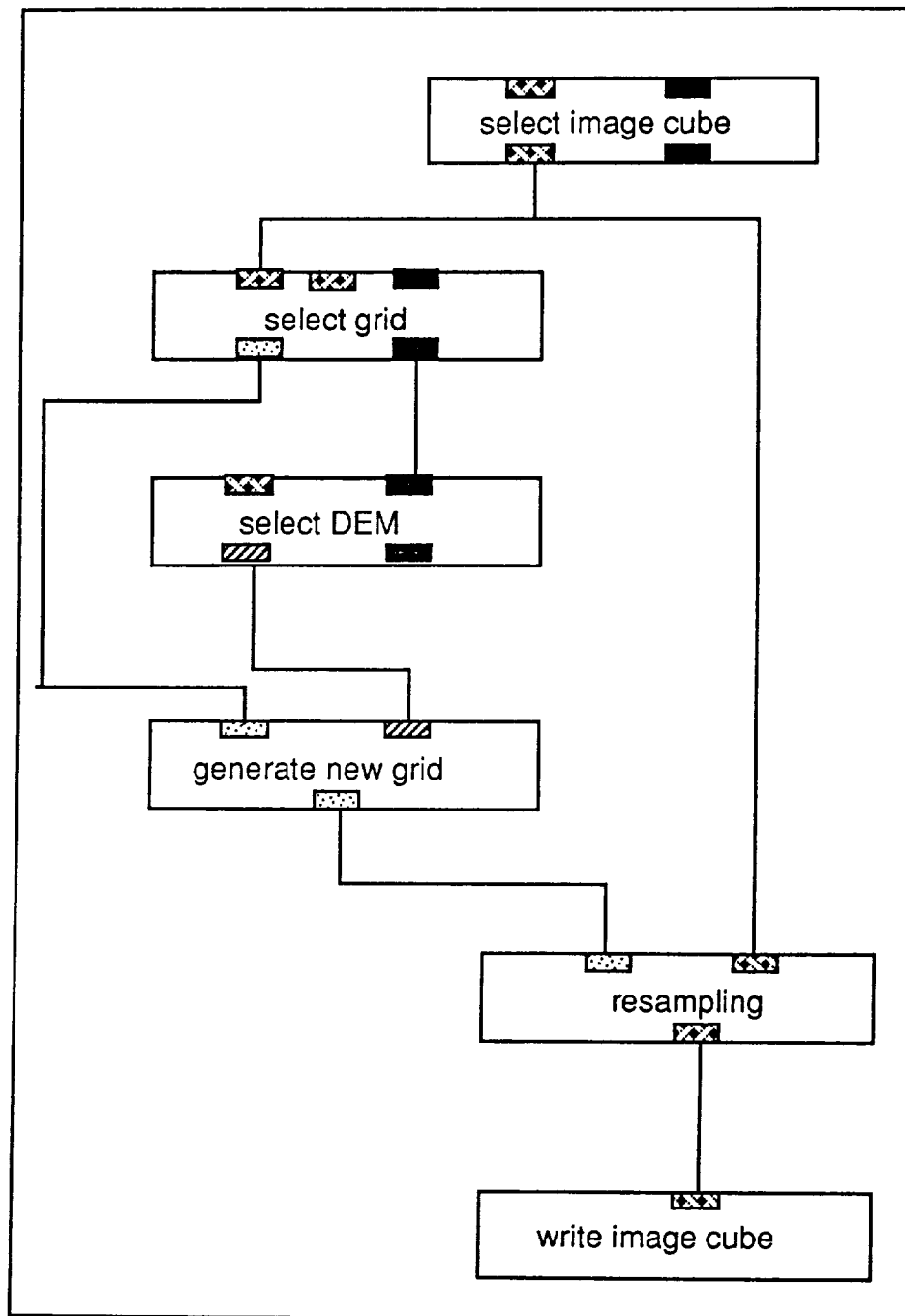


FIGURE 3
RESAMPLING (IMAGE, GRID AND DEM)

Note that the resampling is not via a single set of anchor points. A specific resampling technique will be described in another document. A single concept, namely "resampling" is here shown to be applicable irrespective of sensor or specifics of the geometric situation.

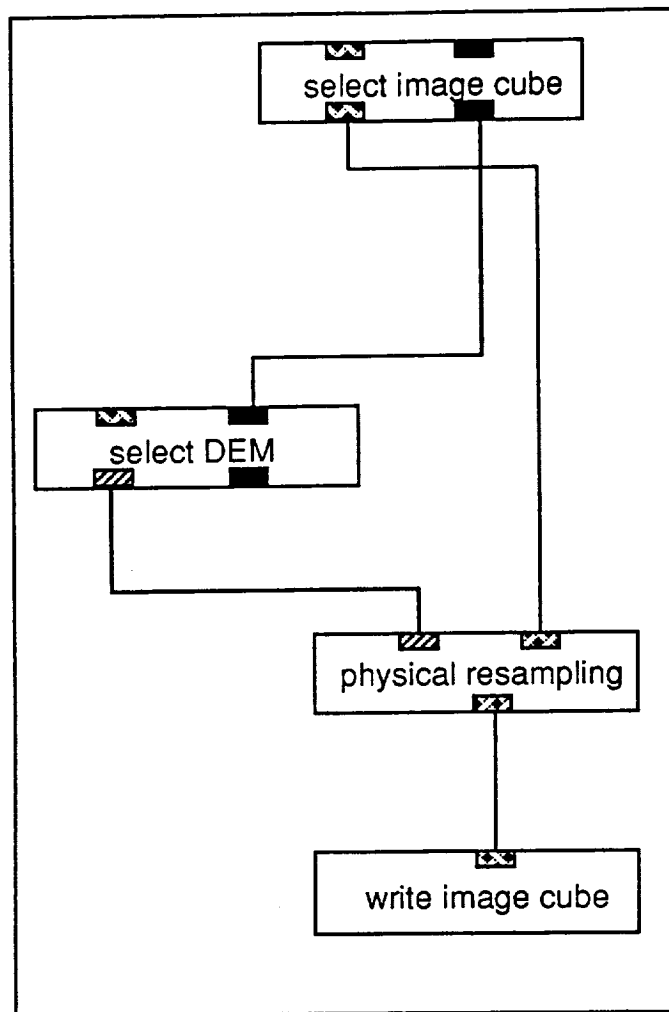


FIGURE 4
RESAMPLING (IMAGE, AUXILIARY DATA AND DEM)

4. GEOMETRIC PROCESSING OF FEATURES

The geometric processing of features allows the user to represent features in different coordinate systems without the burden of acquiring them each time. We defined AVS networks for the following applications:

- (a) Projection of features acquired in an image cube to the ground coordinate system without DEM (see Figure 5). This network can also be used to project features from one image cube to another image cube. It assumes that the mapping between the image cube and the ground, or another image cube, has already been represented by a grid of anchor points. The network is composed of the following modules:

- Select image cube.
- Select grid of anchor points.
- Select features. The image cube name is used to select the features acquired in this particular image cube.
- Project features. Using the grid and the features' image cube coordinates, compute the features' new coordinates.
- Write features.

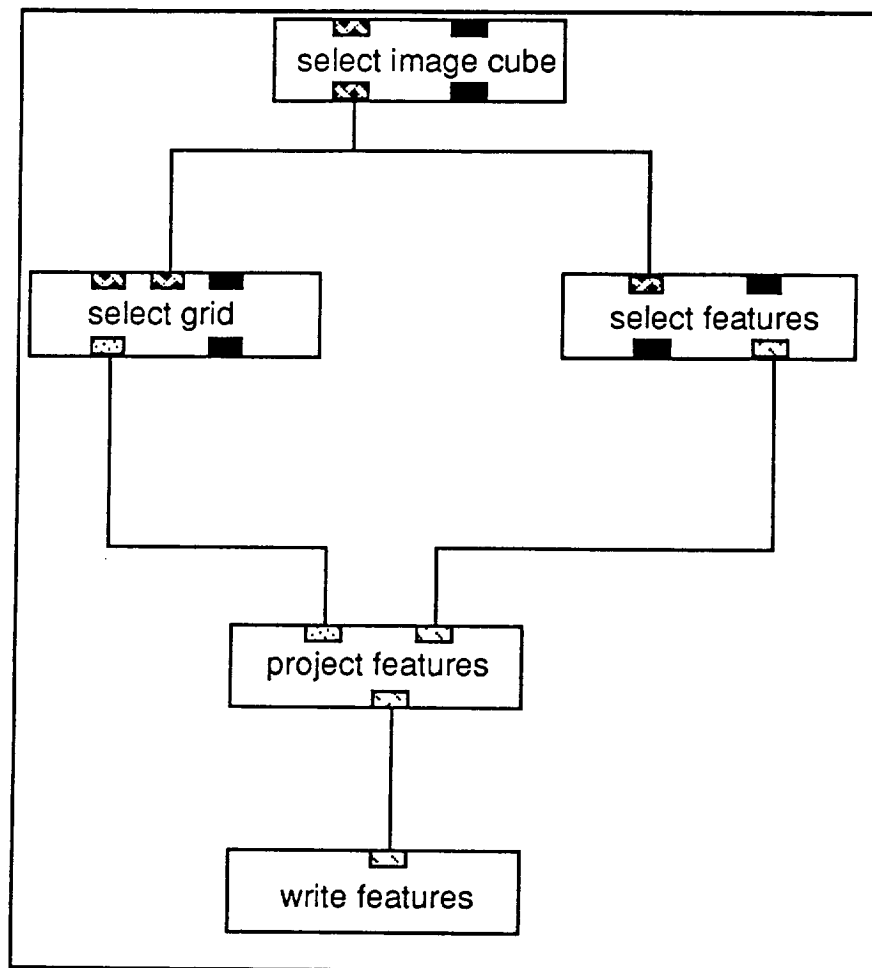


FIGURE 5
GEOMETRIC PROCESSING OF FEATURES
(IMAGE TO GROUND WITHOUT DEM)

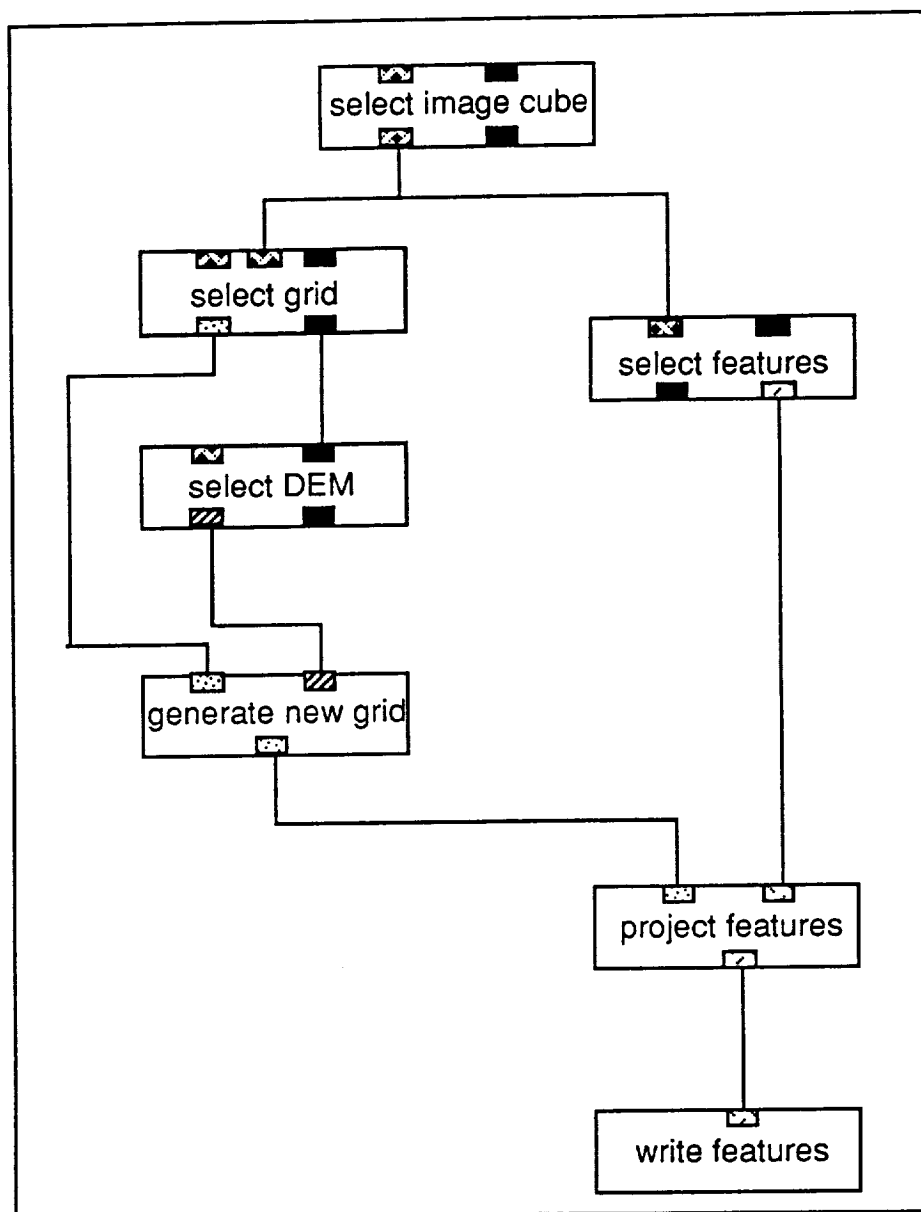


FIGURE 6
GEOMETRIC PROCESSING OF FEATURES
(IMAGE TO GROUND WITH DEM)

- (b) Projection of features acquired in an image cube to the ground coordinate system using a DEM (see Figure 6). This assumes that the mapping between the image cube and the ground has already been established by a grid. This function is composed of the following modules:
- Select image cube.
 - Select grid.
 - Select DEM.

- Generate new grid. This module was described in the resampling with image cube, grid and DEM.
- Select features.
- Project features.
- Write features.

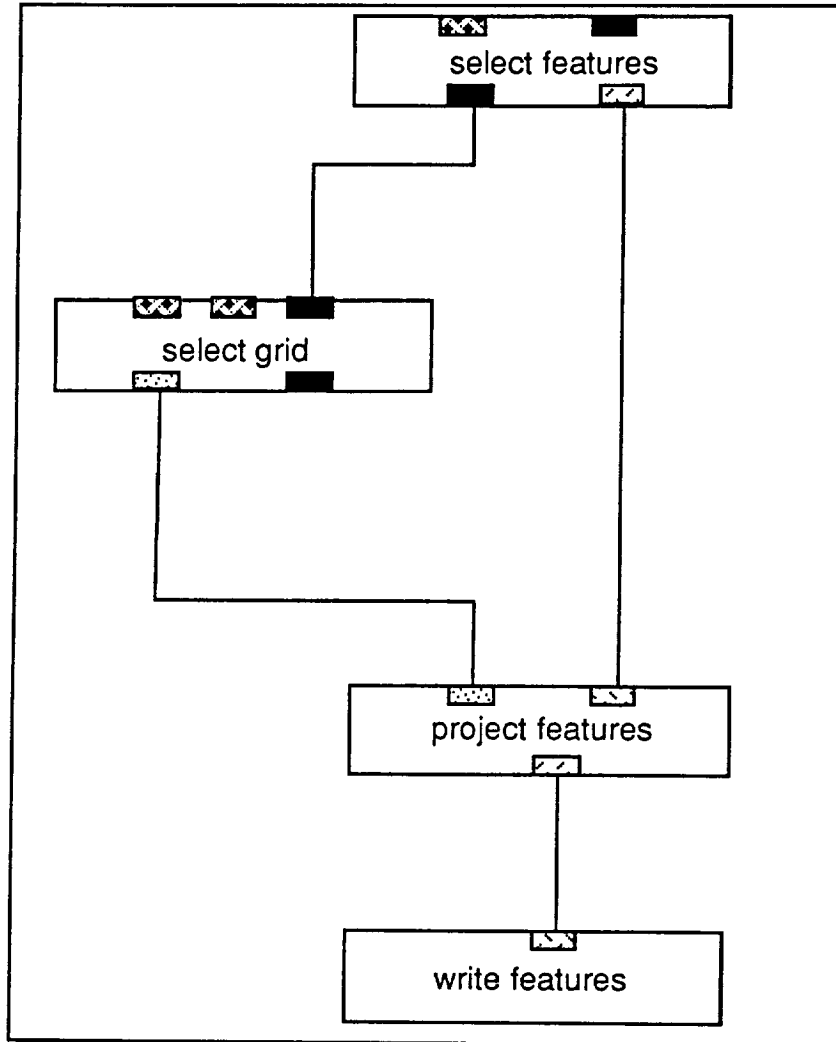


FIGURE 7
GEOMETRIC PROCESSING OF FEATURES
(GROUND TO RAW IMAGE)

- (c) Projection of features, referenced in the ground coordinate system, to an image cube (see Figure 7). It assumes that the mapping between the ground and the image cube has already been represented by a grid. It is composed of the following modules:
- Select features.
 - Select grid. This time the grid selection is done using the 4 corner coordinates of the rectangle containing the features.

Because there is no image cube name as input, the module assumes that the reference is the ground.

- Project features.
- Write features.

(d) Projection of features, in the ground coordinate system, to a geocoded image cube (see Figure 8). This demonstrates the use of the general AVS network to accomplish a simple coordinate translation. This is composed of the following modules:

- Select features.
- Select image cube. This time the image is selected using the 4 corner coordinates of the rectangle containing the feature. Because there is no input image cube, this module assumes that one deals with the ground coordinate system.
- Generate grid. This grid will contain the exact mapping between the cartographic/geographic coordinates and the image coordinates using the image cube parameters.
- Project features.
- Write features.

(e) Projection of features, referenced in a geocoded image cube, to the ground (see Figure 9). This is necessary because features referenced in an image cube have only image cube coordinates. The network is composed of the following modules:

- Select image cube.
- Select features.
- Generate grid. This grid will contain the exact mapping between the image coordinates and the cartographic/geographic coordinates using the image cube parameters.
- Project features.
- Write features.

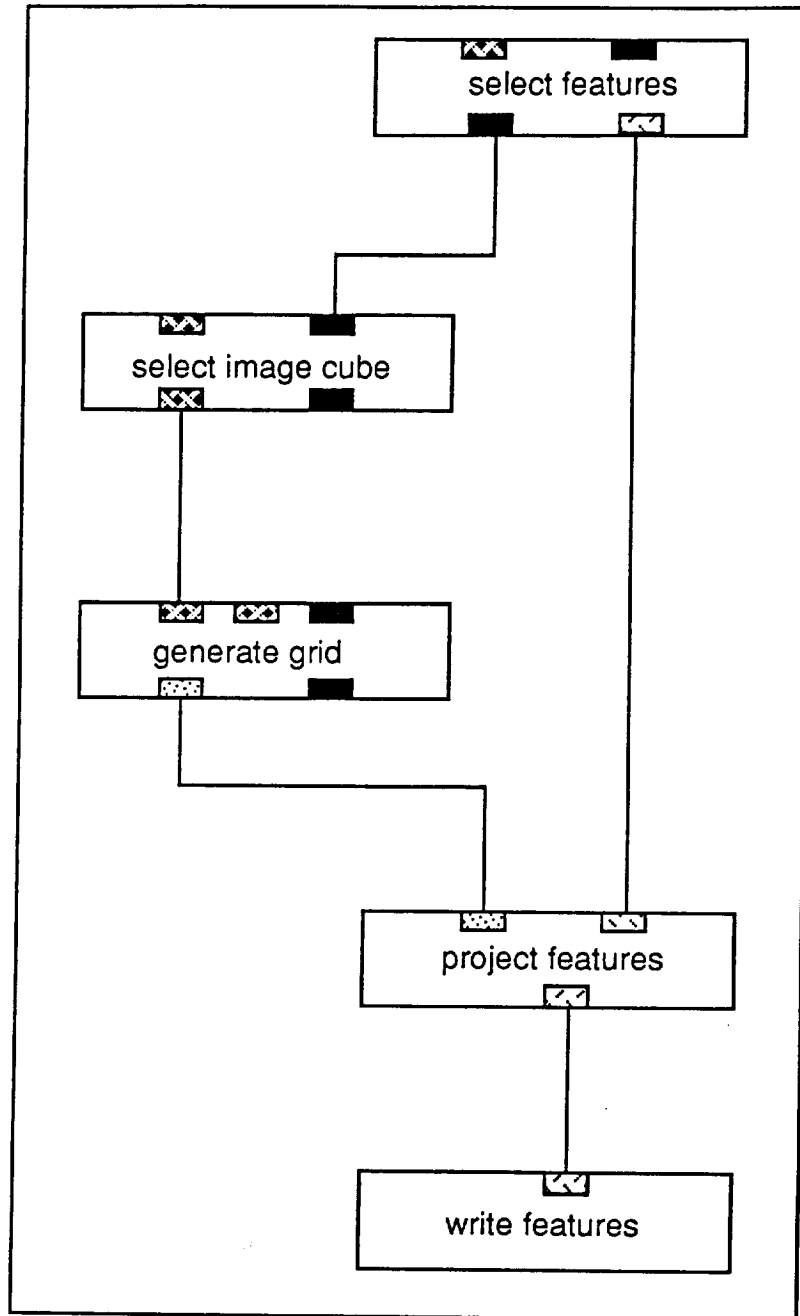


FIGURE 8
GEOMETRIC PROCESSING OF FEATURES
(GROUND TO GEOCODED IMAGE)

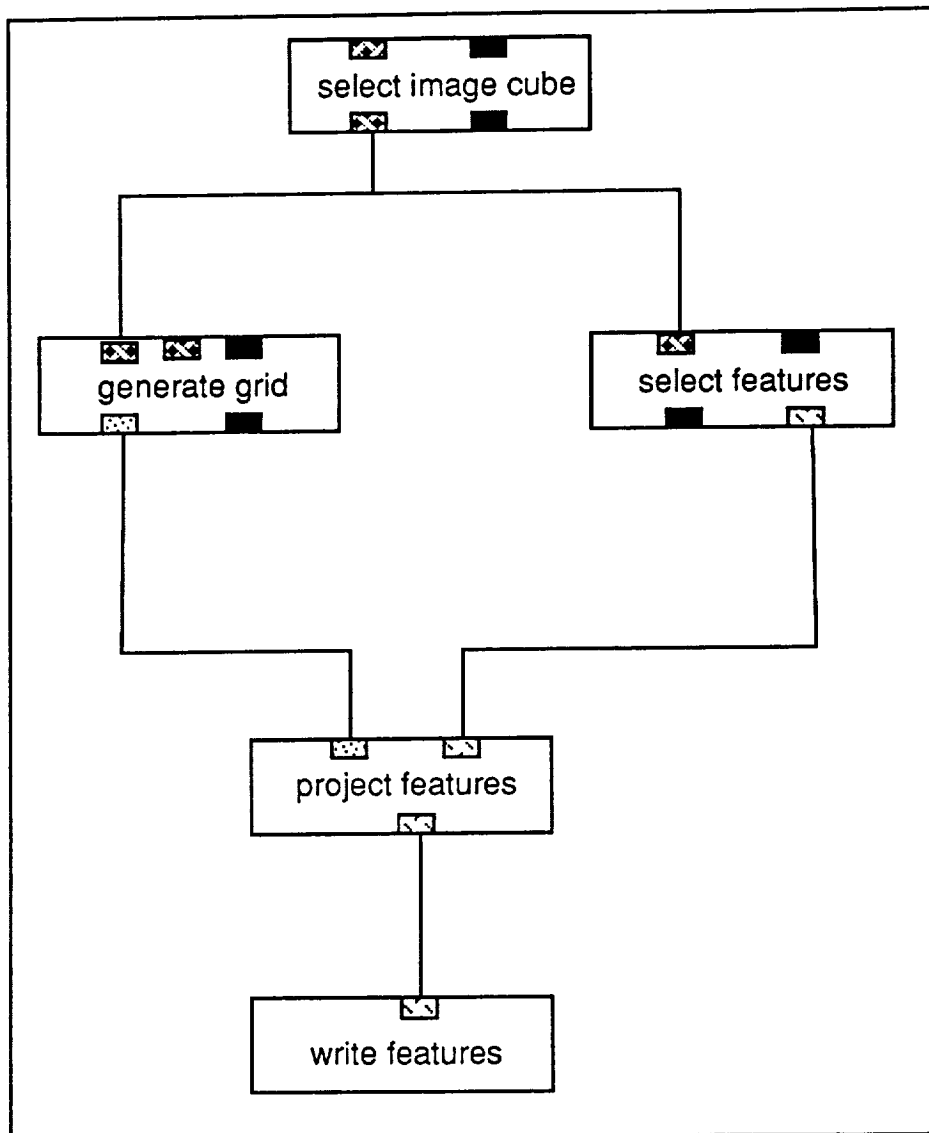


FIGURE 9
GEOMETRIC PROCESSING OF FEATURES
(GEOCODED IMAGE TO GROUND)

5. IMAGE CUBE GENERATION

Image cube generation allows the user to merge into the same object different images coming from different sources. We worked with AVS networks for the following applications:

- (a) Image cube generation when the coordinate reference is the ground and there exists no image cube covering the area of interest (see Figure 10). It is composed of the following modules:
 - Select coordinate system. The reference can be the ground or an image cube.

- Select image cube. The absence of an input image cube implies that we use this module to select geocoded images; each image is a special case of an image cube.
- Select DEM.
- Select features.
- Fill image cube. Using the previously defined area, this module serves to transfer individually selected images into the image cube.
- Reorder image cube. This module reorders the layers inside the image cube.
- Write image cube.

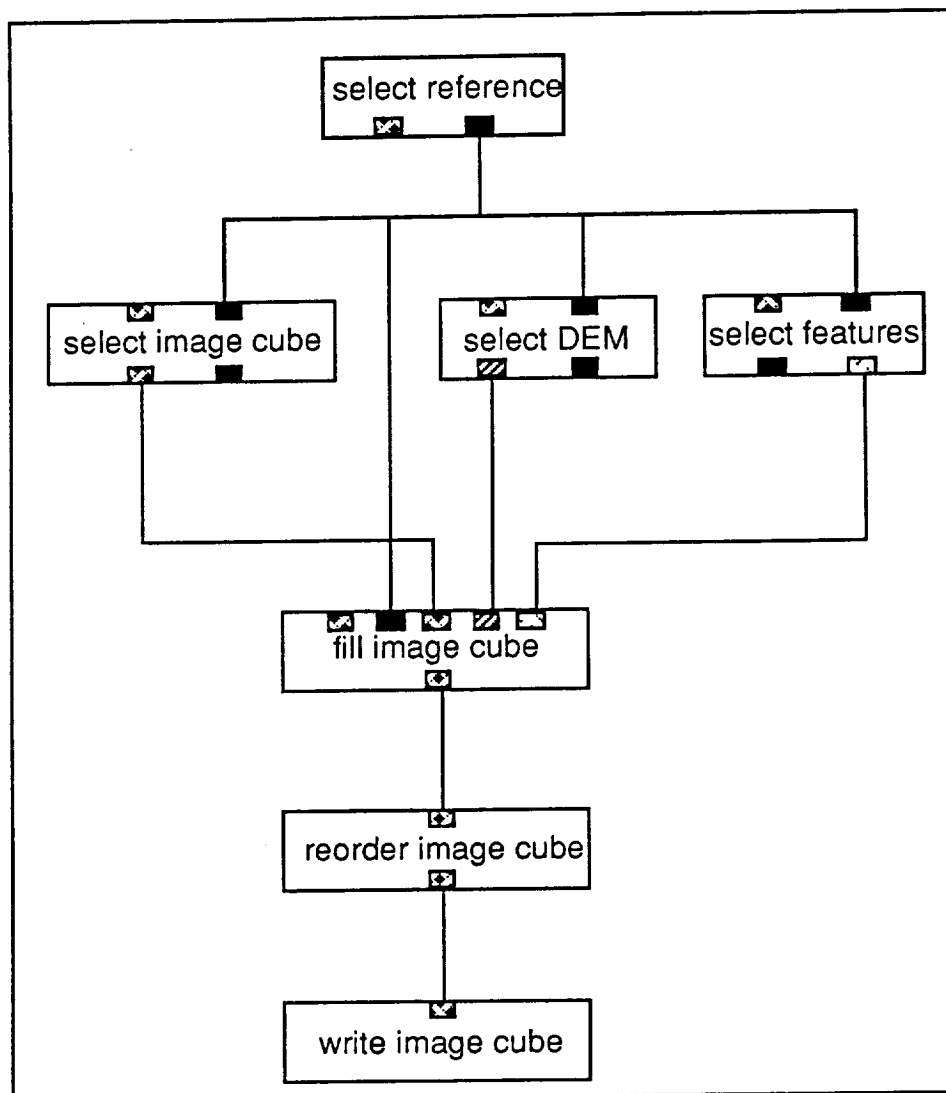


FIGURE 10
IMAGE CUBE GENERATION
(GROUND AND NEW IMAGE CUBE)

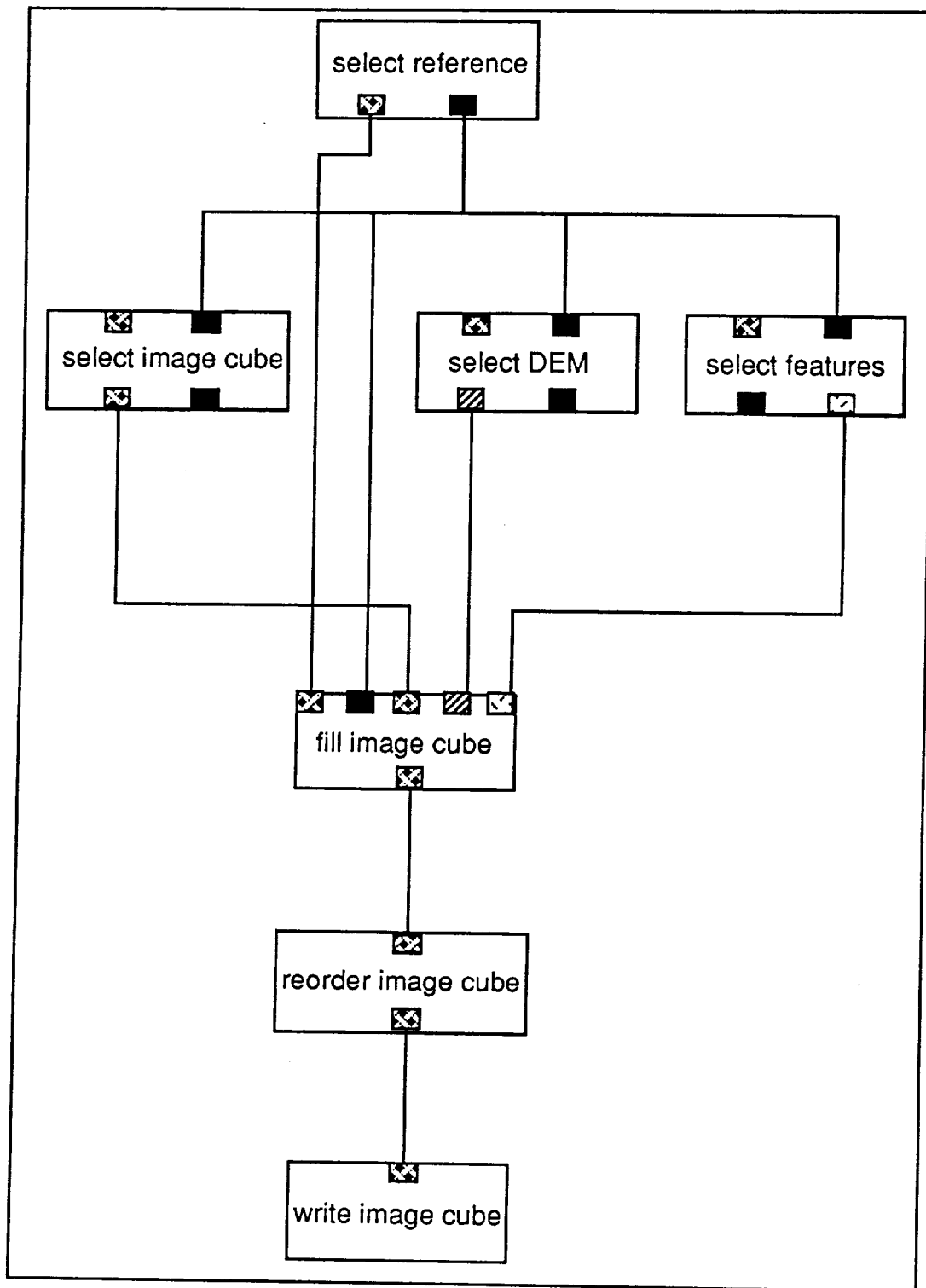


FIGURE 11
IMAGE CUBE GENERATION
(GROUND AND EXISTING IMAGE CUBE)

- (b) Image cube generation when the coordinate system is the ground and there exists already an image cube covering the area of interest (see Figure 11). It is composed of the following modules:
- Select coordinate system reference. An image cube is selected and the user may define an area included in the selected image cube.
 - Select image cube.
 - Select DEM.
 - Select features.
 - Fill image cube. The first image cube input in the diagram indicates that the area was defined using this particular image cube and the output may be considered as an update of this image cube.
 - Reorder image cube.
 - Write image cube.
- (c) Image cube generation when the coordinate system is a non-geocoded image cube (see Figure 12). It is composed of the following modules:
- Select coordinate system. An image cube is selected and the user may define an area included in the selected image cube.
 - Select image cube. The selection is done using the input image cube name as well as the 4 corner image cube coordinates defining the area of interest.
 - Select DEM.
 - Select features.
 - Fill image cube. The first image cube input in the diagram indicates that the area was defined using this particular image cube and the output may be considered as an update of this image cube.
 - Reorder image cube.
 - Write image cube.

6. CONCLUSION

We described AVS-networks to demonstrate the possibility of using AVS in the EOS Workstation. Once modules are defined and written, the user can rearrange the networks and customize their use to particular needs. We will write modules transforming EOS Workstation objects to AVS data types, therefore giving the user the possibility to fully use the power of AVS in configuring specific application solutions.

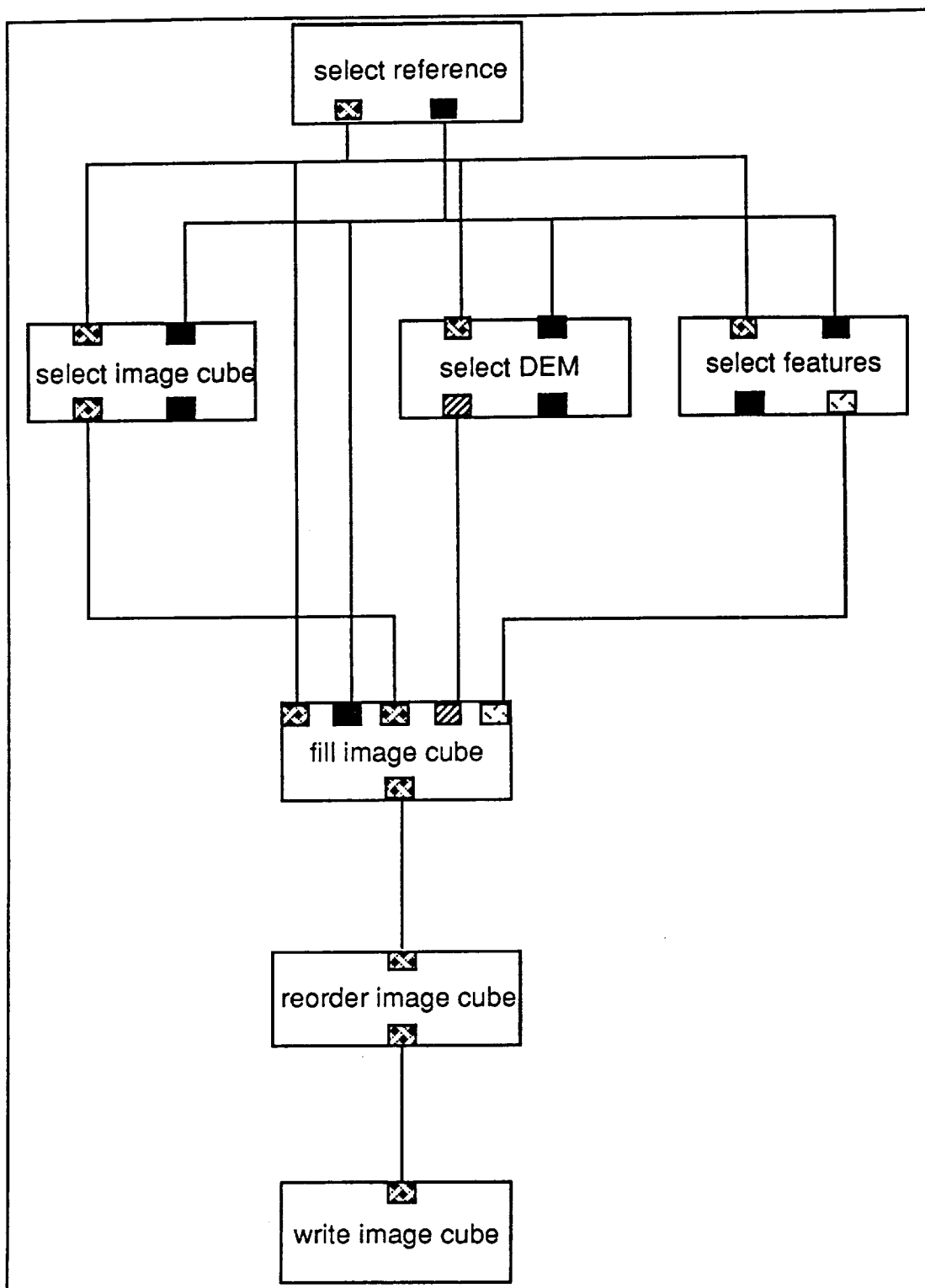


FIGURE 12
IMAGE CUBE GENERATION
(IMAGE IS THE REFERENCE)

ATTACHMENT K

BAND REDUCTION AND SPECTRAL PROTOTYPES

Vexcel Corporation

EOS Workstation

Report on:

Band Reduction Analysis and Design

and

Introduction of the *Prototype Signature* Data Element

K. Maurice

August 6, 1990

Report on Band Reduction and Spectral Prototypes

This document summarizes recent analysis and design efforts for the **dimension reduction** portion of Vexcel's EOS Workstation. This effort has been undertaken as an independent step, primarily in response to overlapping requirements between Vexcel's EOS and HIRIS projects. This document **only** addresses aspects of band reduction as related to the EOS project. This document references the EOS Analysis document of 6/15/90.

The use of **prototype signatures** is also introduced in this document. This new data dictionary element in the EOS system specification is being added to support the input and output of signatures that come directly or indirectly from the data itself. It was decided that users will almost certainly want this feature and therefore it would be a required addition to the system capability. This type is introduced and described in this document from the point of view of analysis but functions that manipulate it are not addressed from a design standpoint.

Also briefly discussed is the evaluation of AMOEBA as an example of a complex user-defined software application that can be hosted on the EOS Workstation.

Document Organization

First the basic context within which the dimension reduction process and prototype signature data element reside is outlined. Then the basic issues involved and the decisions made about them are detailed. *Specification* and *design* information is then given in the form of DFD's, user option trees and textual description (analysis), and structure charts (design). Finally, implementation details, where relevant, are provided.

It is anticipated that analysis and design work outlined in this document will be reviewed and when validated, merged into EOS specification and design documents.

EOS Subsystem Context

Dimension reduction falls under the general category of **image cube visualization**, which in turn is analysed in the context of Image Cube Analysis (see EOS System Specification). In general band reduction is to provide a mechanism for reducing the number of image bands from n to m while retaining the original multi-band image structure. Given the large number of AVIRIS bands as well as other sensor data with multi-temporal aspect, this feature was seen as essential to the workstation functionality.

Report on Band Reduction and Spectral Prototypes

The new data element, *prototype signature*, is used also within image cube analysis, but in the context of **spectral or polarization analysis**. While it is interesting to compare laboratory or other user imported signatures with signatures taken or derived from image data it was deemed important to also *maintain* these data derived signatures. Additionally a user may want to export these derived signatures for use outside the workstation. These prototype signatures must be related to images and their features for further investigation. This necessitates the need for image and feature information to be associated with prototype signatures, whereas they were not for spectral or polarization signatures, which may not be inherently related to image data.

Possibilities for the use of prototype signatures in other areas of the EOS Workstation (such as Geometric Processing, etc) have not been addressed.

Band Reduction Issues

Background

Originally, the intension of this process was to provide a mechanism for creating and applying a transformation from n-dimensional to m-dimensional imagery. How this was to be achieved was an open issue but two methods were immediately identified as candidates:

i. using the classical *principle components transformation* to produce a mapping from n to n space where the output n images were orthogonal to each other. Ideally most of the *detail* in the first few output bands comprises *most* of the information content of the original data. The user has the responsibility of selecting m of these bands ($m \leq n$) to implicitly form a reduction process.

ii. investigate the automatic construction of a transformation from n to m space that preserves the pair-wise differences between prototype classes in the n-band and m-band images. This might amount to generating an $M \times N$ matrix that maps N-vectors into M-vectors while minimizing the sum of the difference between pairs of N and M-space prototype pixel norms.

Current Directions

The first solution is one that has been implemented in a variety of ways by researchers and appears to be at least one straightforward method that can be realized for the EOS effort. While problems do exist with this technique (such as performance and accuracy for large band reductions, use of efficient numerical techniques, difficulties in interpretation and color assignment, etc.), they can be dealt with; variations on this method include MNF and other PCA-based tech-

Report on Band Reduction and Spectral Prototypes

niques that can be implemented as user-functions.

The second approach requires more thought. Finding the matrix described above, using norms, it turns out, is a non-trivial problem in non-linear optimization that must deal with finding a global minimum amidst a very large number of local minima. This is equivalent to finding a solution to a system with a very large number of unknowns, a computationally exhaustive process.

Initially, it was believed that the public domain program *AMOEB*A, authored by Jack Bryant at Texas A&M, somehow addressed the above problem in providing a general-purpose multi-dimensional image clustering and band reduction process. We had hoped to be able to extract the dimension reduction code from the program and further, to extend it to generate results in m space (not 3-space), $m \geq 3$. The problems with this approach are the following:

- 1.) The program uses a PCA trained on "prototype pair differences" for starting the cluster analysis and another PCA combined with a color stabilizing algorithm to reduce n -space to 3-space. It appears that although the reduction is extractable from the larger program it is really nothing more than a PCA (with the addition of the color mapper). The solution of the more general problem of minimizing the "objective" function described above was actually abandoned by the author until a better method for solving "nonlinear optimization" problems could be found. The results of this earlier work can be found in [Bryant and Guseman, 1979. Distance preserving linear feature selection, *Pattern Recognition* 11:347-352]. The substitute was PCA with (apparently) the justification that measurement space under this transformation is still preserved.
- 2.) The subroutines performing the PCA and color assignment (REDD, REDDIS) are difficult to understand. In addition the theory itself of the color mapping is hard to grasp. The code is the result of incremental development over a number of years, with much attention to optimization for particular hardware and software configurations (ie. PC-based disk drives, VAX VMS operating system), and disk and memory requirements. Portability would be a major concern here.

While it may not be advantageous to pursue the use of this part of *AMOEB*A it is useful to at least include the reduction process by prototypes as a possible function in the workstation. So a place will be left open for minimization based on the objective function defined above (or some other measure) in the system specification document, but no design work will be done at this time. Nor will any extraction of *AMOEB*A code take place.

Report on Band Reduction and Spectral Prototypes

Use of *AMOEB*A as an Example of a User Program under EOSW

While this program is grand in scope and functionality it is unclear as to its usefulness for the EOS project. The reasons for this include those listed above under band reduction as well as the following:

- 1.) The clustering-classification appears to be somewhat image dependent - that is it works very well for broad-band multi-spectral images like Landsat and MSS but perhaps not so well for higher spatial or spectral resolution data (aircraft, or AVIRIS). Even more crucial is the overall inability to define precisely what kind of data it *is* most appropriate for.
- 2.) The large number of input parameters makes the program difficult to understand and use. There are scores of interactive paths that can be taken, making the use of the system often difficult. It is not clear what the correct parameters would be for EOS Workstation data.
- 3.) The overall time required to fully understand, port and host this program would be too great for the EOS effort given the hard constraints on time and resources weighed against the utility derived from such a program.

The conclusion then is that although *AMOEB*A should **not** be used as a primary example of a complex user application hosted under EOS, it should be considered as a possible future or secondary example of one. If a new version came out based on Unix or including some other useful modification it might at that time become a candidate.

Prototype Signature Issues

The primary issue relating to this new data element is the desire to relate these new data derived signatures to image features maintained in the system. For example a user would outline a geological area of interest in the imagery and then perhaps store the "average" signature associated with this area for later use. Conversely the user might have such a signature, derived previously from image data, that is to be compared with data in other imagery. Features in the imagery matching this prototype could be located by the system.

In order to relate signatures and features the *labels* and *codes* of features will be used as identifiers. Codes in general correspond to broad, hierarchical categories of natural origin (ie. hydrology, forestry, etc.); these are encoded with specialized extensions to further subdivide the categories (ie. types of water, forests, etc.); then a label, possibly describing the geographical name of an entity is used.

Report on Band Reduction and Spectral Prototypes

The main point about the development of this new type is the decision of whether to store the {x,y} feature point locations with the prototype which would essentially duplicate the database. At this time we are planning to NOT include this information but rather to allow these coordinates to be extracted from the feature database through the use of *coding/label* keys. Another more general issue to be addressed is how to handle user-defined codes. Currently the solution being considered is to have a *miscellaneous* code (ie. 00).

The use of prototype signatures is introduced in this document but it is expected that there may be some further elaboration or expansion of its definition and use during the final systems analysis and design phases of the project.

In general, the functionality added by the use of prototypes is identical to that of *spectral signatures* except that the workstation can generate signatures for later retrieval whereas spectral signatures can only be input to the system.

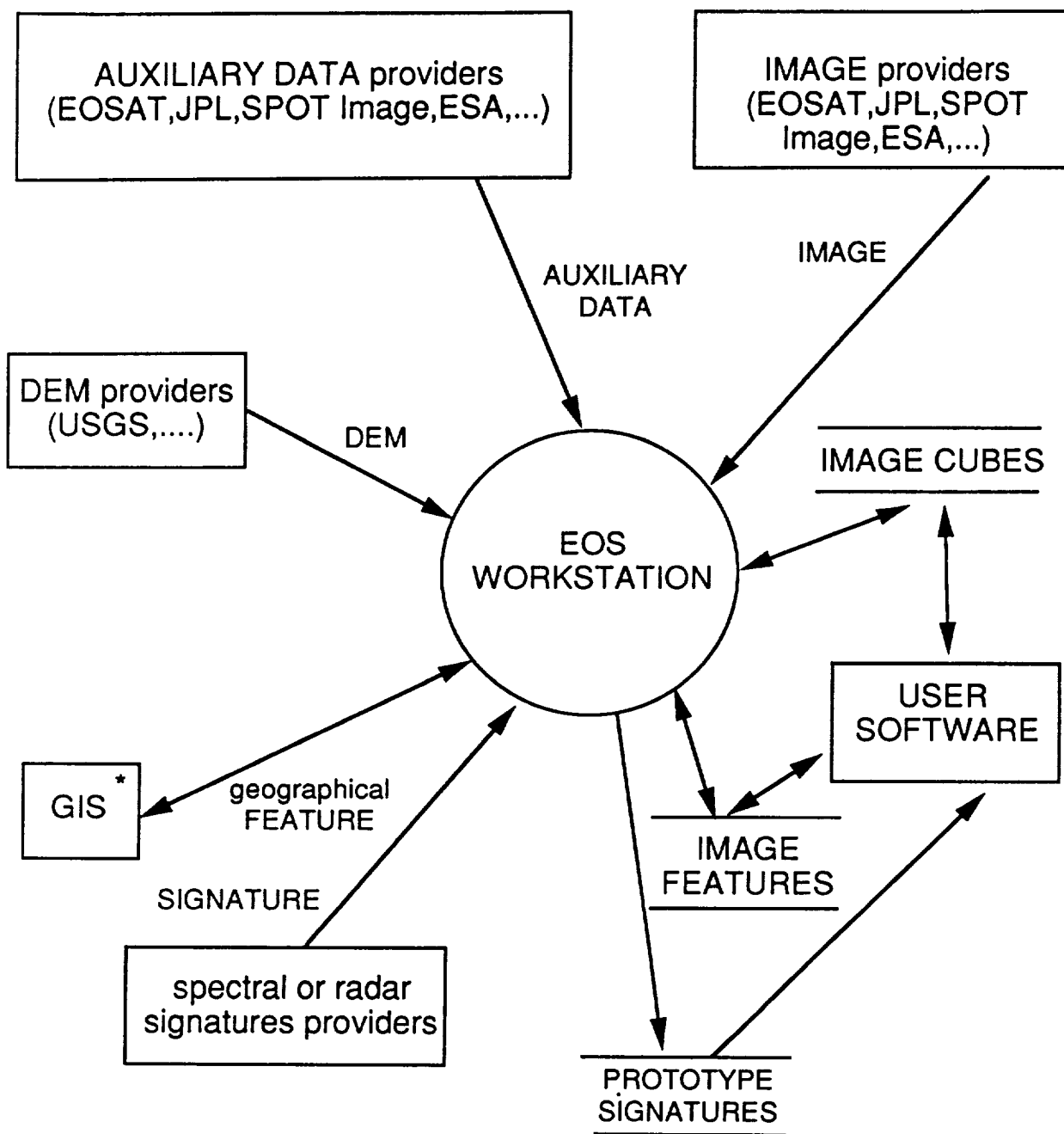
Update to Analysis Based on Prototype Signature Addition

Data Flow Diagrams

Prototypes may be *read* by the user. Therefore the context diagram and Level 0 diagrams must be updated to reflect this. These diagrams follow. The Data Dictionary definition for the new type precedes the diagram.

PROTOTYPE SIGNATURE	=	prototype name + image cube name + image name+{band name+DN}NSP + (1{FEATURE NATURE}NF) * DN digital number NSP number of signature points NF number of features*
---------------------	---	--

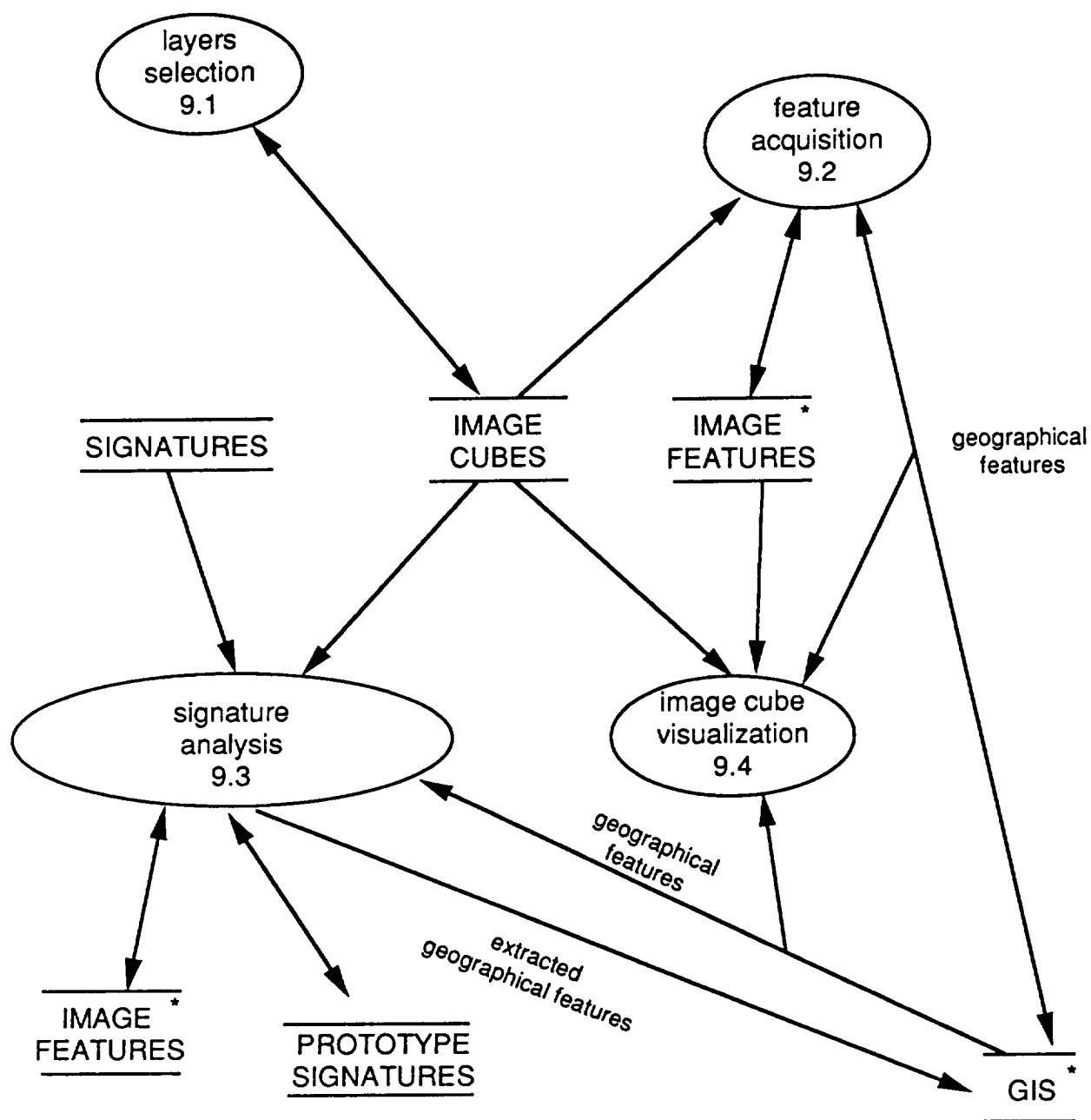
NOTE: the implication of the above definition is that only one image cube can be associated with a given prototype at a time. If multiple cubes are involved it is up to the user to either assign a new prototype name for a new cube or different layers for different images within a cube. Also, no specific image information is given since it is assumed that this information can be derived from the image cube name and layer name(s). Also in the context diagram that follows for the stores accessed by the user we are assuming that the contents of these are equivalent to those found internally on the Level 0 and other diagrams but their *formats* are different. This is an implementation detail and it is left to the design phase to provide variations of user-defined input/output formats.



EOS WORKSTATION
CONTEXT DIAGRAM

Report on Band Reduction and Spectral Prototypes

The diagram following shows the new EOS Workstation Analysis Figure 9 resulting from the addition of prototype signatures into the specification. The DFD reflects the fact that now prototype signature can be read by and written from the spectral or polarization analysis process (now called signature analysis).



EOS WORKSTATION DATA FLOW DIAGRAMS

FIGURE 9: IMAGE CUBE ANALYSIS

Report on Band Reduction and Spectral Prototypes

User Options Diagrams Updates

The following show the new diagrams for the User Options Figures 11, 12, and a new diagram 13. These are for the new band reduction options and prototype signature types.

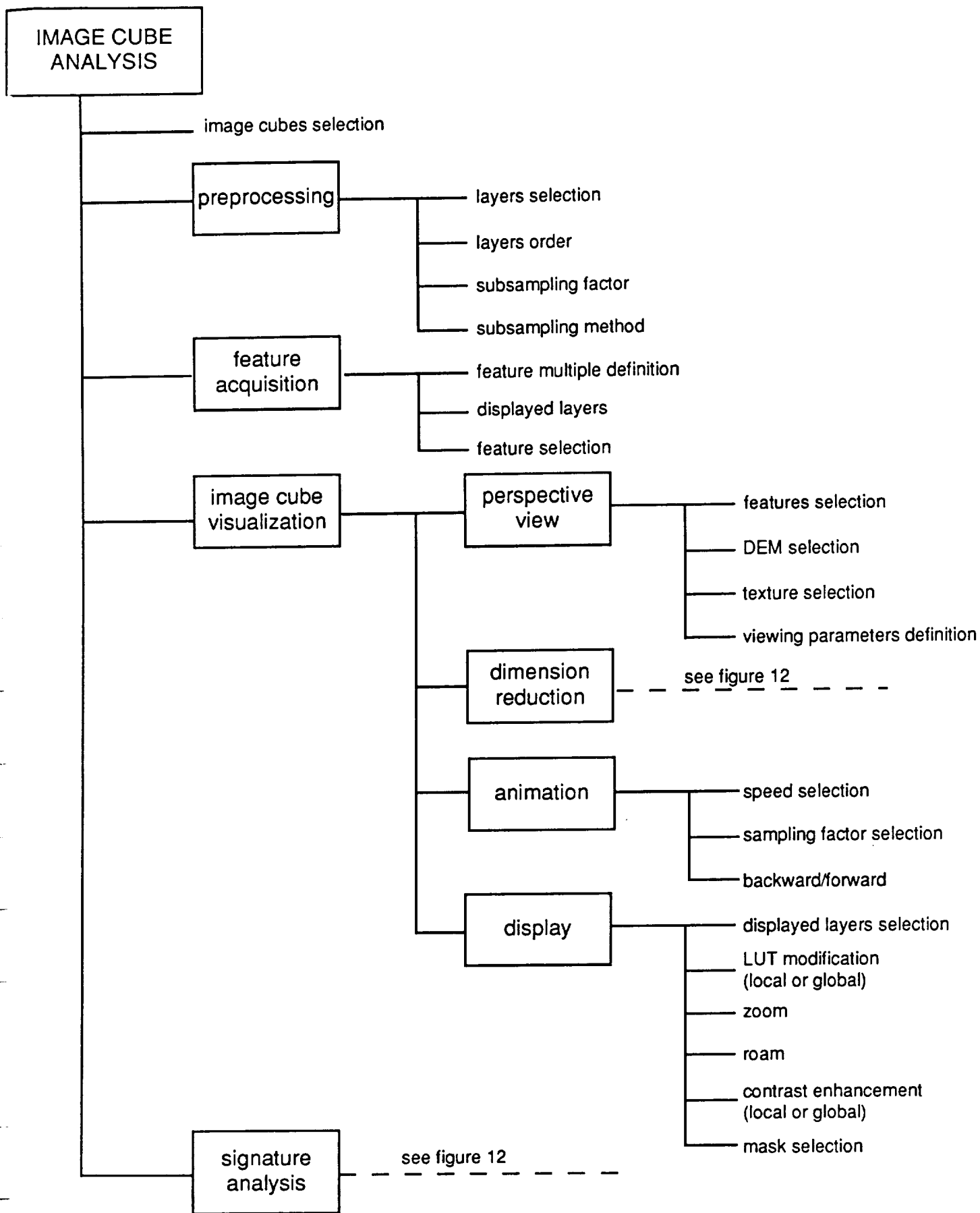


FIGURE 11
USER OPTIONS: IMAGE CUBE ANALYSIS

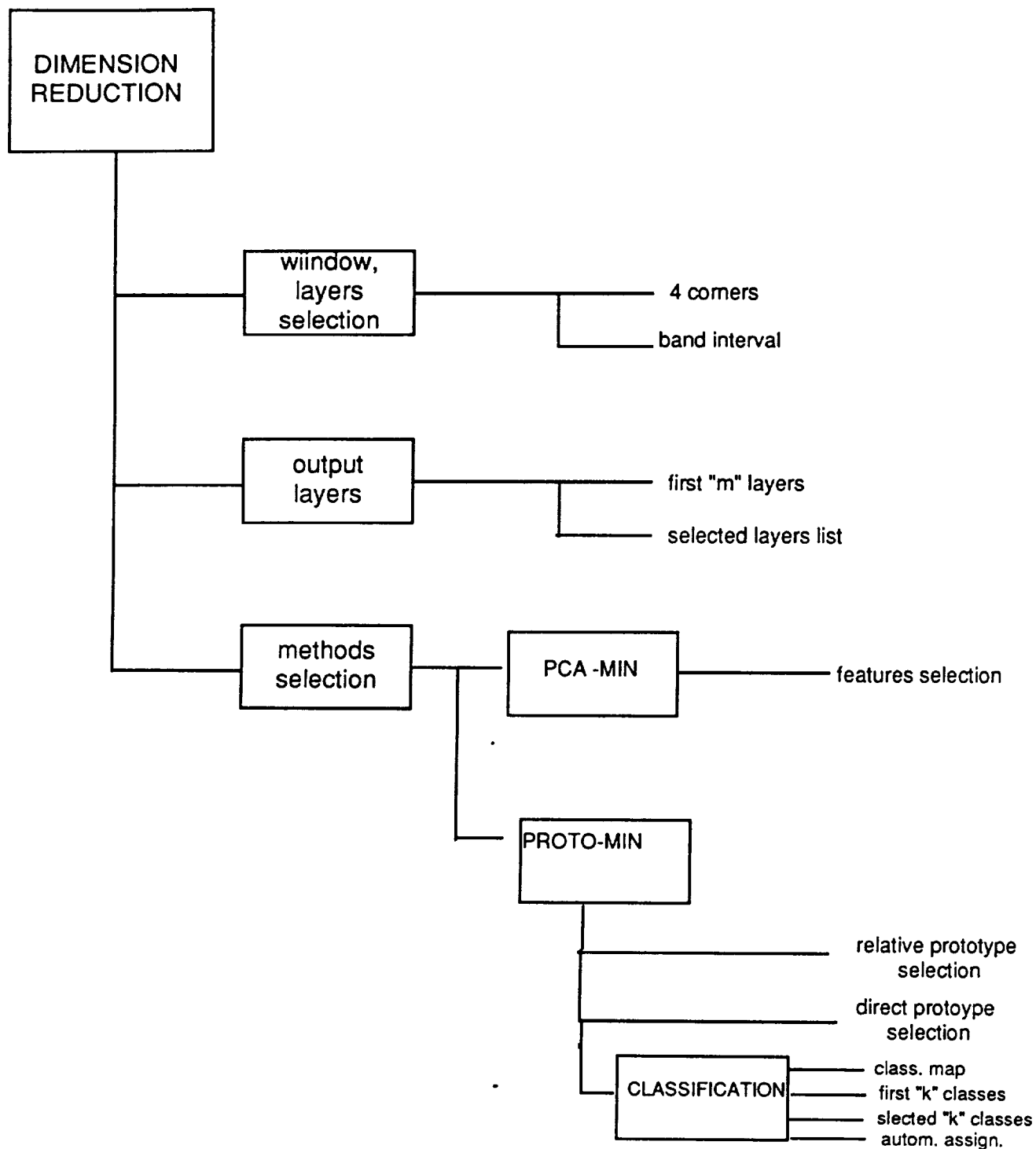


FIGURE 12
USER OPTIONS: IMAGE CUBE ANALYSIS
(DIMENSION REDUCTION)

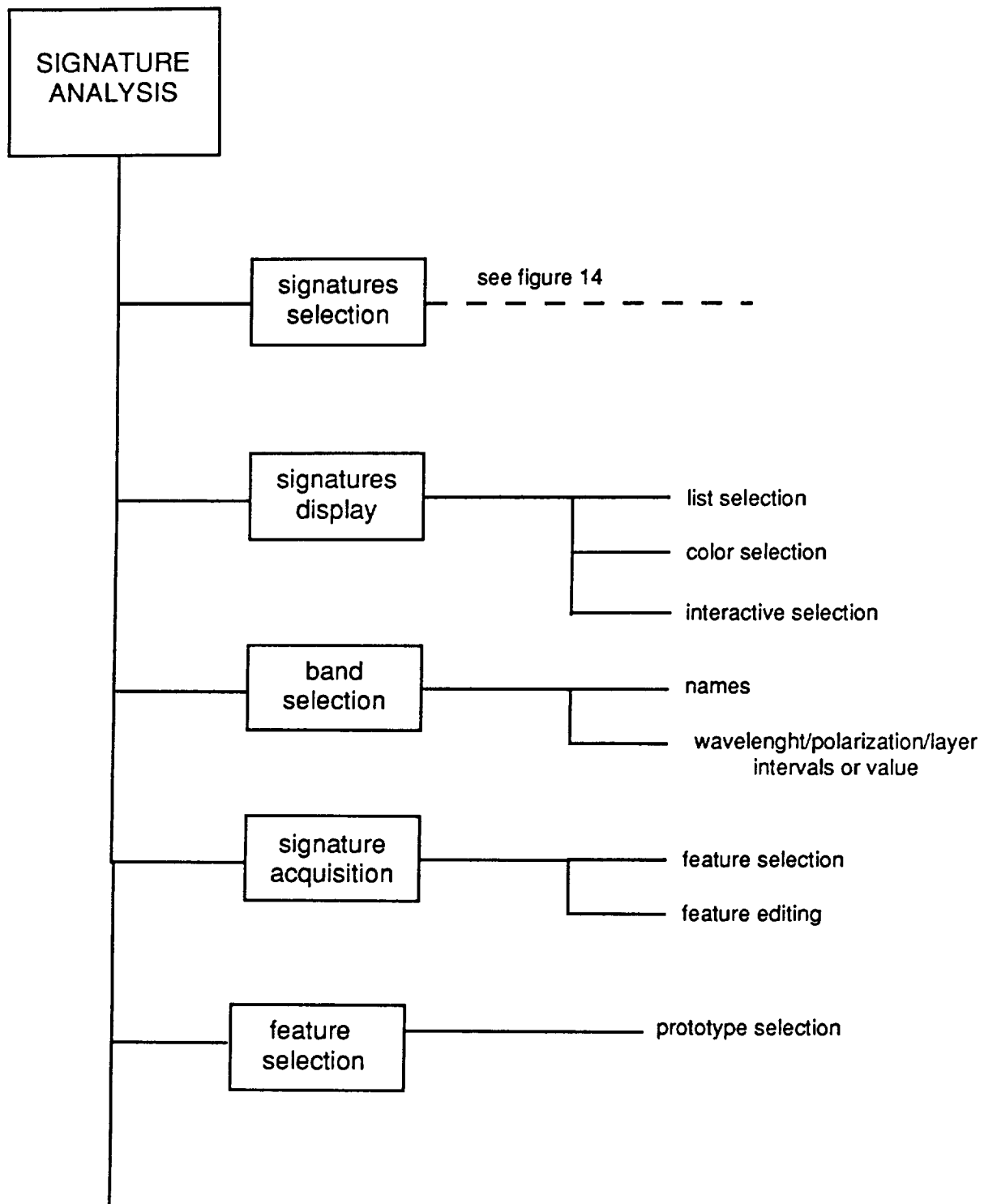


FIGURE 13
USER OPTIONS: IMAGE CUBE ANALYSIS
(SIGNATURE ANALYSIS)

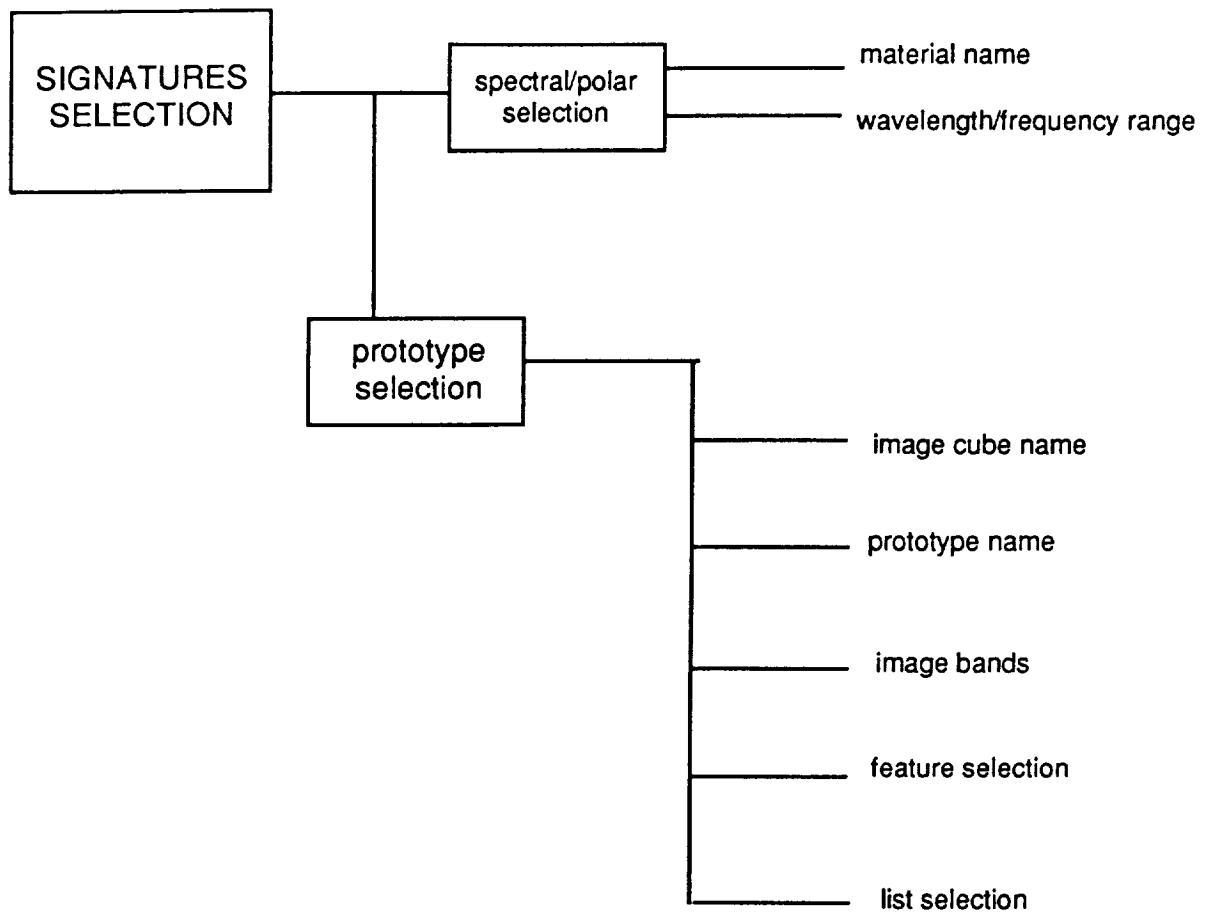


FIGURE 14
USER OPTIONS: SIGNATURE ANALYSIS
(SIGNATURES SELECTION)

Changes to the FUNCTIONAL DESCRIPTION

These changes would be incorporated as is into the Functional Description section of the EOS Analysis document by section number. It is assumed that existing descriptions will remain and that the following will be added.

Section 6.3 Image Cube Visualizatio / Dimension Reduction

The user can reduce the number of bands from n to m . This can be done for an entire image or for sub-areas or feature defined areas if PCA minimization is selected. For PCA minimization the user can either choose the first m principle components or else specify a list of specific output components. For Prototype minimization the user can choose automated or direct methods of prototype input which may or may not be based on classification. For PCA minimization, features may be selected for exclusive processing and would thus require the use of a mask on output.

Section 6.4 Spectral or Polarization Analysis (Signature Analysis)

Prototype signature analysis is restricted to one image (multiple bands). The user can:

- Select one of several signatures. The user can select a signature by image cube name or from a prototype list. Signatures can be selected by features with respect to an image also. Prototypes may further be selected by prototype name or by prototypes defined in the bands of an image.
- Display prototype signatures. The user can sequentially display selected signatures or display selected signatures at the same time with different colors. The choice is made by stopping the sequential display or by indicating it in the window containing the selections.
- Compare prototype with spectral/radar signatures. This is identical to selection and display of prototypes or spectral/radar signatures except selections are made across both types.
- Select image bands associated with derived prototypes. This function is similar to that for spectral/radar signatures. New image cubes can be generated based on these intervals.
- Display features associated with prototypes. The user can select one or more features related to prototypes. These features can be overlaid onto an image corresponding to one of the selected prototype bands for the image.

Report on Band Reduction and Spectral Prototypes

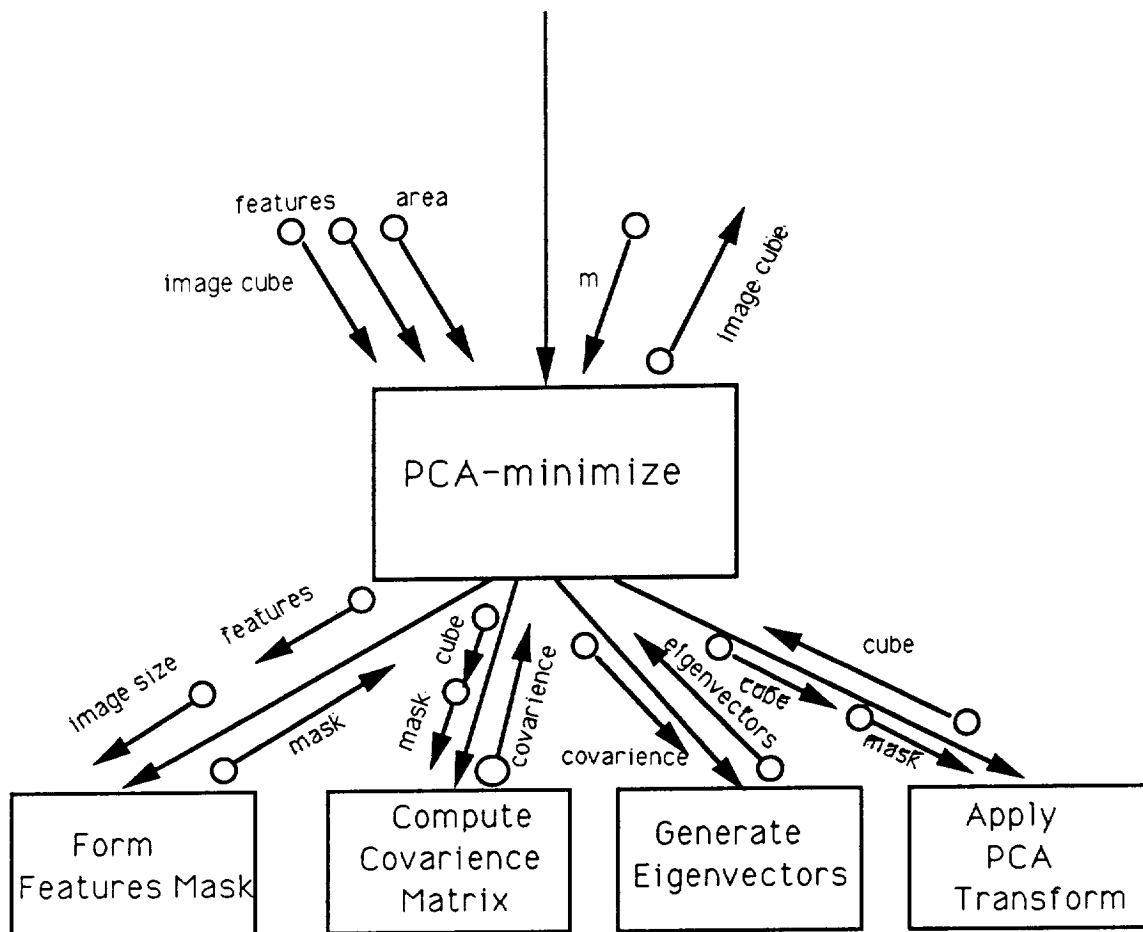
- Display, acquire and select prototypes: this is the same as that defined in the Analysis document except that "pixel signatures" are now prototype signatures and that these stored signatures are retained by the system and can be accessed by user external to the system. Additionally the user may specify the image band intervals for the desired prototype.

- Acquire feature signatures. Based on previously acquired features the user can define and store prototypes. Prototypes can be directly acquired based on a single pixel from a feature or derived from feature pixels through averaging. The user must first select and/or display the associated feature for the target image. Additionally the user can define new features in the manner described in the Analysis document (ie. polygonally); the feature is stored in the features list as well as the signature lists and and/or means. Features are labelled and stored according to an image cube and its layers.

Design Documentation

This section of the document contains a structure chart for the PCA-minimization function. This chart reflects the basic needs for this function and is meant as a starting point for further design elaboration. No detailed design elements are included here such as data specifications and I/O descriptions, algorithmic detail, etc. Nor are error conditions listed. The following section discusses an implementation of this function that can be modified to conform to the design. A short discussion of these required modification is presented.

It is anticipated that the above design issues will be formally resolved during the primary design stage for the EOS effort when the majority of system functions are elaborated.



Structure Charts: PCA-minimization

Existing PCA-minimization Function Implementation Issues

Currently there is a C program in existence that will generate and apply the principle components transformation to a cube. A cube in this case is just a band-sequential array of images in byte format. The program will take all three dimensions as parameters:

pcavolume #bands #lines #pixels <ret>

By default it reads from a file called **rendvol.dat** and writes to **pcavolume.dat**. It can optionally generate AVS *volume* files. At this time it tries to allocate a 3D array defined by the dimensions so it will probably not work for very large dimensions. The program should run on most Unix systems.

At this time there needs to be an added check of the form $A*x = \lambda*x$, where λ is an eigenvalue, x is an eigenvector and A an arbitrary matrix. There is a check that verifies that the sum of the original covariance matrix diagonal elements is equal to the sum of the eigenvalues but that is not enough.

The program calls C functions for

- . memory allocation
- . covariance matrix computation
- . eigenvector/eigenvalue generation (Numerical recipes Jacobian function)
- . sorting of eigenvalues/eigenvectors (NRC)

It should be noted that the Jacobian method is said to work well for $N \leq 20$ but is slower for larger. A QR method should be used there. NRC has functions for both.

Also, there is no generation of any masks as required by feature-based processing. Nor is there any PCA based on a features list. It is expected that these can be easily added to the current implementation. A linear scaling is performed to map from PCA space to display space (0 ... 255).